BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi

Issue 131 | July 2023 | magpi.cc

The official Raspberry Pi magazine

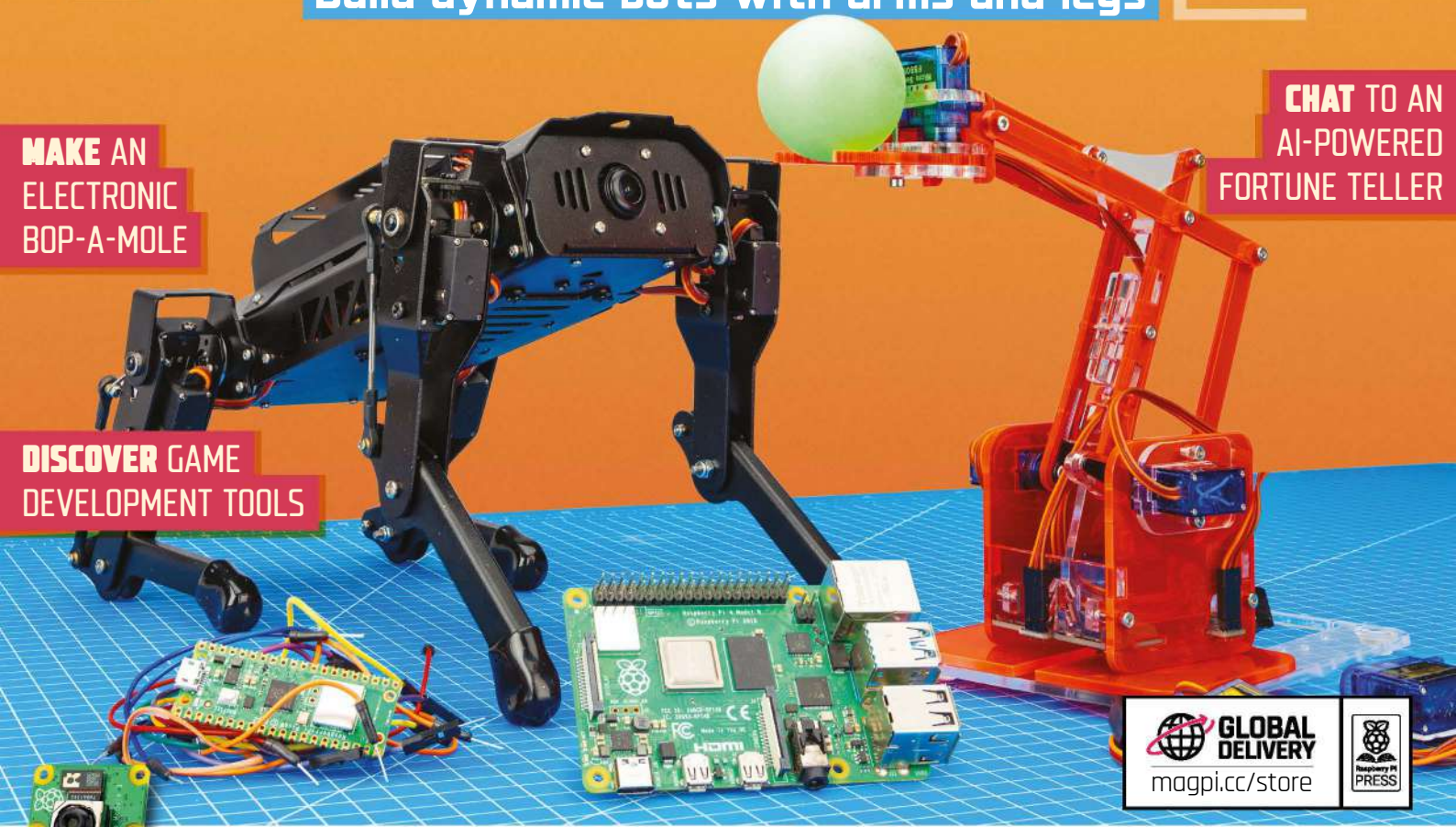
Python Robots

Build dynamic bots with arms and legs

MAKE AN ELECTRONIC BOP-A-MOLE

CHAT TO AN AI-POWERED FORTUNE TELLER

DISCOVER GAME DEVELOPMENT TOOLS

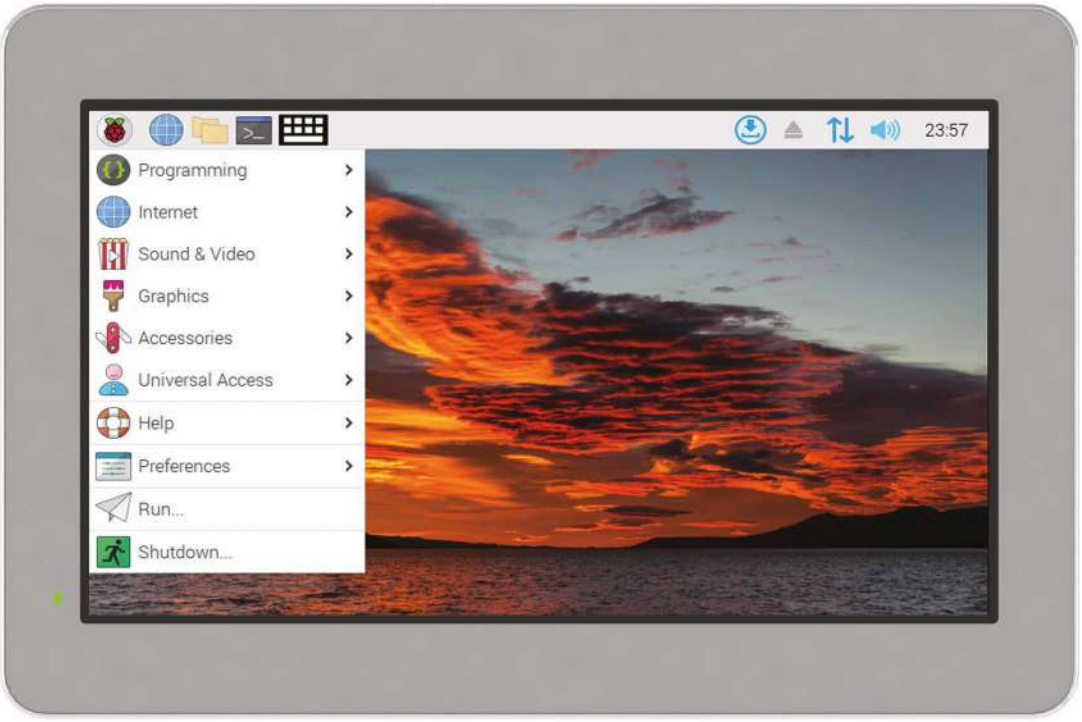


GLOBAL DELIVERY
magpi.cc/store
Raspberry Pi PRESS

CREATE A **RASPBERRY PI WILDLIFE CAMERA**

UL LISTED

Industrial Raspberry Pi



ComfilePi

The ComfilePi is a touch panel PC designed with high-tolerant components and no moving parts for industrial applications. It features a water-resistant front panel, touchscreen, color LCD (available in various sizes), RS-232, RS-485, Ethernet, USB, I2C, SPI, digital IO, battery-backed RTC (real-time clock), and piezo buzzer.

Use the rear-panel 40-pin GPIO header to expand its features and capabilities with additional I/O boards. The ComfilePi is UL Listed and employs Raspberry Pi Compute Module.

POWERED BY Raspberry Pi

Visit www.comfiletech.com

© copyright COMFILE Technology, Inc. ALL RIGHTS RESERVED

COMFILE TECHNOLOGY

WELCOME
to The MagPi 131

We love animals and robots here at *The MagPi*. There's lots of similarity: both need training, feeding, and nudging with frustration and love.

Lately, we've tested some incredible bots with arms, legs, and limbs. All are controlled by Raspberry Pi and powered by servo motors. So, we thought it time to limber up and herd them together into a giant Python Robots feature (**page 28**).

If real-life animals are more your thing, check out Rob's feature on capturing wildlife with Raspberry Pi (**page 66**). Setting up camera traps to snap wildlife is a great hobby that lets you take only photos without scaring critters.

This month's magazine ticks a lot of boxes for me. But there's much more to it than animals and robotics. Flick over the page and take a look at the contents.

Lucy Hattersley Editor

EDITOR

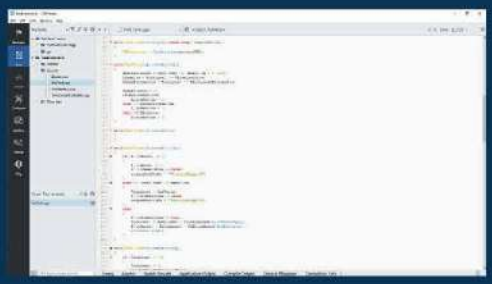
Lucy Hattersley

Lucy is editor of *The MagPi* and was snapped at quite a few times by dogs as a pestering child.

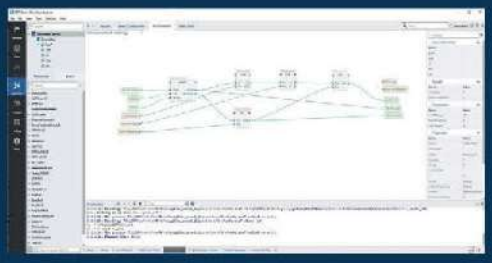
magpi.cc

GET A RASPBERRY PI PICO W WITH A SUBSCRIPTION! PAGE 22

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Python Robots
Build dynamic bots with arms and legs
CREATE A RASPBERRY PI WILDLIFE CAME

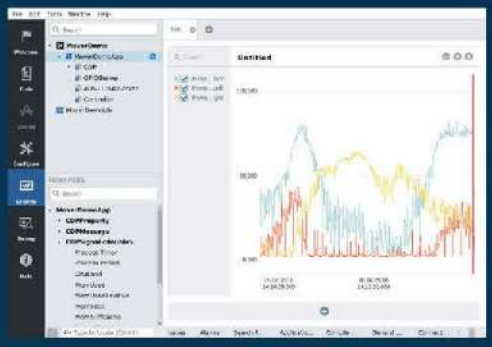


Full-Code



No-Code / Configure

Design



Analyze

now with a

No-Code WEB UI Designer

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

CDP Studio, a great software development tool for your home projects. Build systems for Raspberry Pi, use C++ or NoCode programming, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create beautiful user interfaces. Built for industrial control system development, **FREE for home projects**.

cdpstudio.com
Tel: +47 990 80 900 • info@cdptech.com
CDP Technologies AS // Hundsværgata 8, 6008 Ålesund, Norway

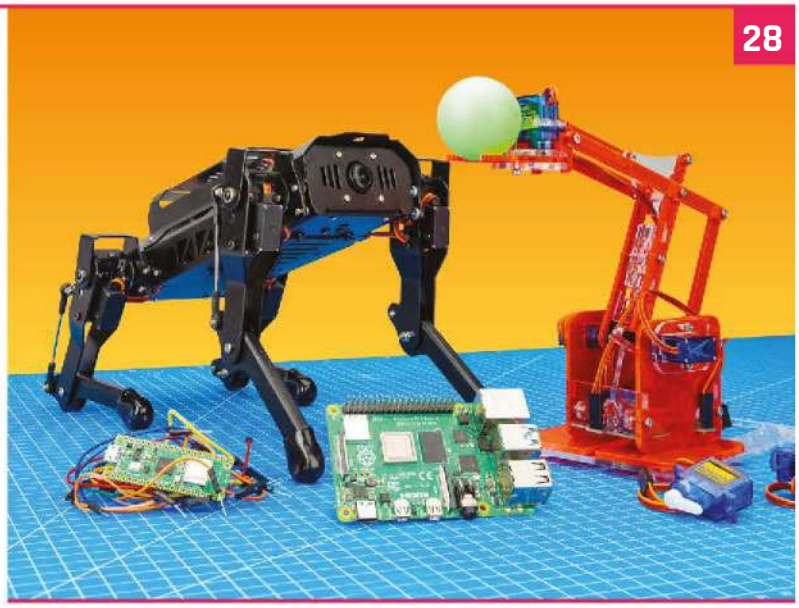
CDPStudio

Contents

Issue 131 > July 2023

Cover Feature

28 Python Robots



Regulars

24 Case Study: Sisyphus
90 Your Letters
97 Next Month
98 The Final Word

Project Showcases

08 Trainbot
12 Ohsillyscope
14 Olga The Fortune Teller
18 Chonky Pocket
20 GPS Pothole Tracker

Chonky Pocket

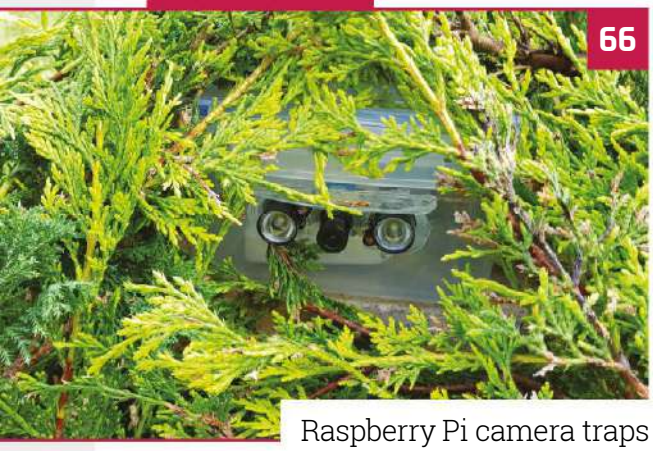
Olga The Fortune Teller

The MagPi is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The MagPi, c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Tutorials

40 Game engines for Raspberry Pi
48 Build a Mole Bop game
52 Taking video on Camera Module
58 Rad your Ride
62 Create interactive stream lights

The Big Feature



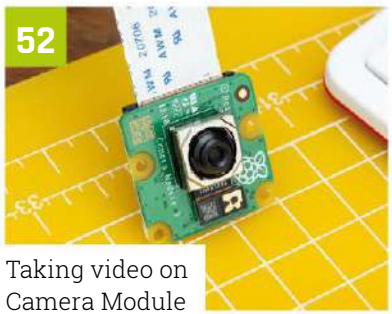
Raspberry Pi camera traps

Reviews

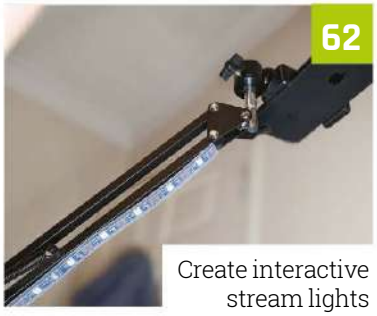
72 PuppyPi
74 u-maker Box
76 Inky Impression 7.3 inch
77 WuKong 2040
79 Badger 2040 W
80 10 amazing cases
82 Learn Assembly

Community

84 Alistair MacDonald interview
86 This Month in Raspberry Pi



Taking video on Camera Module



Create interactive stream lights

Badger 2040 W

Alistair MacDonald interview

WIN 1 OF 2 **ARGON POD SETS**

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

NEW PRODUCTS
START HERE

Looking for the latest products? Look no further. With over 400,000 new products stocked, we've got your electronic component and automation needs covered.

Find it at **digikey.co.uk/new** or call 0800 587 0991.

Digi-Key ELECTRONICS

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2023 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

Trainbot/OnlyTrains

A grandstand view led a Swiss hobbyist to record passing trains using Raspberry Pi's camera, learns **Rosie Hattersley**

MAKER

Jonathan Müller

Software engineer Jonathan enjoys challenging code projects and has been using Raspberry Pi for nearly a decade.

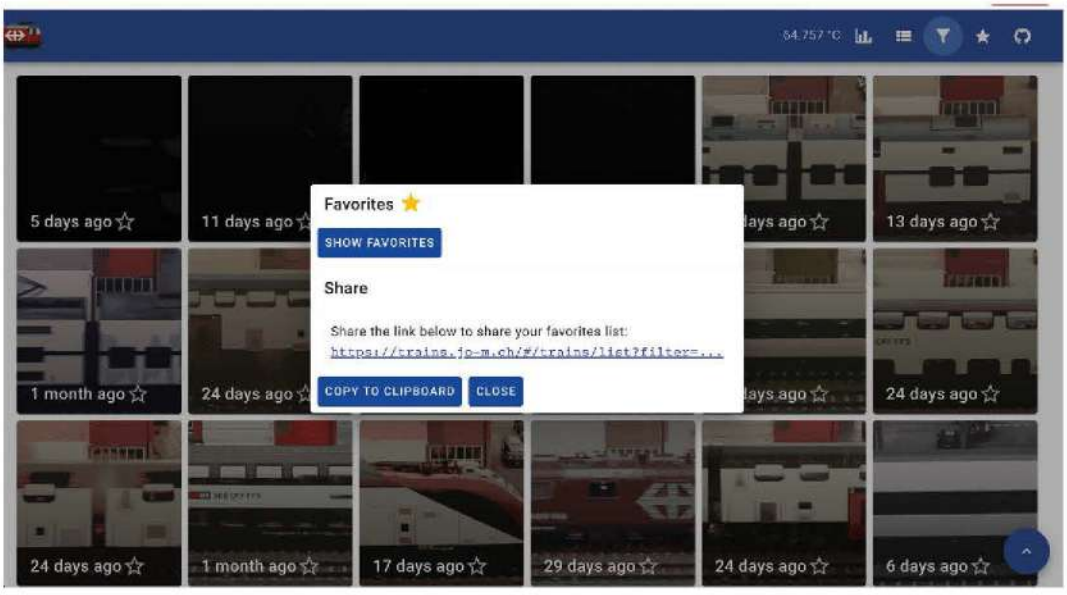
trains.jo-m.ch

Living in sight of a railway has proved inspirational for many: Suggs from Madness; Rod Stewart, your writer's Dad! It was also the catalyst for Trainbot, an automated photography project triggered by the engines and carriages running past the window of maker Jonathan Müller in Zurich, which stitches shots together to appear as complete trains. Acknowledging the geeky but passionate hobby of trainspotting, he created a website on which to view the resulting photos and named it OnlyTrains.

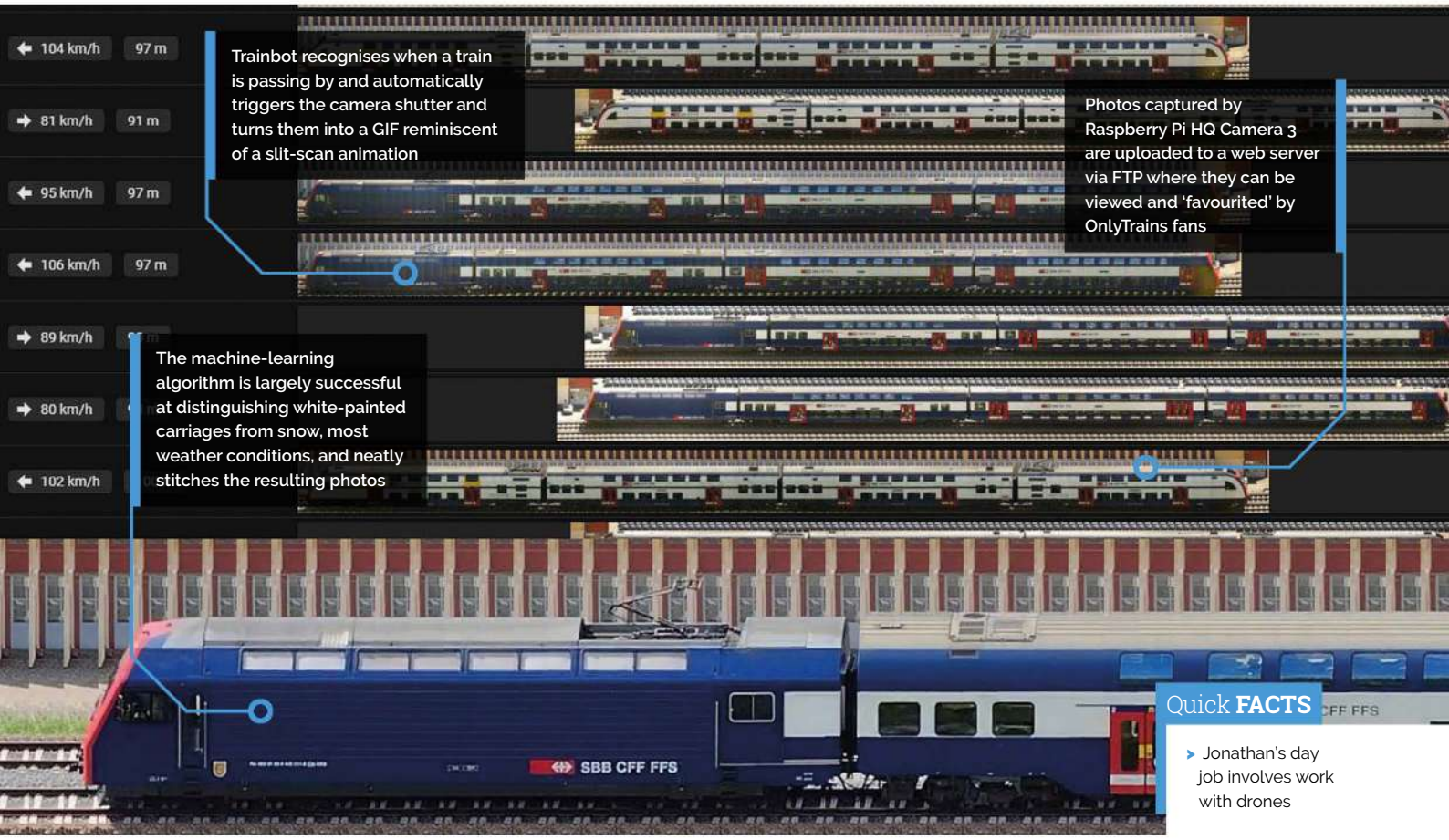
Room with a view

Jonathan first came across Raspberry Pi when he was still an electrical engineering student back in 2014. He promptly put his first Model A Raspberry Pi to good use controlling the door to a co-working space at the ETH Entrepreneur Club where he was a member, using Slack (**magpi.cc/dooropener**). There isn't much call for the computer vision skills that were part of his degree course, but Jonathan enjoys creating projects that exploit his technical knowledge.

When Jonathan moved into his apartment two years ago, he quickly realised that the unique view it gave him of the trains going by would make for an ideal project subject. "Inspired by the slit-scan finisher photographs from bike races, I then came up with the idea of creating pictures with similar aesthetics – but of trains instead of bikes, and a cheap camera instead of specialist hardware," he explains. Slit-scan photography



Each image is made up of small patches, each of which has been processed individually in order to create a single final composite image


104 km/h
97 m
81 km/h
91 m
95 km/h
97 m
106 km/h
97 m
89 km/h
80 km/h
102 km/h
Trainbot recognises when a train is passing by and automatically triggers the camera shutter and turns them into a GIF reminiscent of a slit-scan animation
Photos captured by Raspberry Pi HQ Camera 3 are uploaded to a web server via FTP where they can be viewed and 'favourited' by OnlyTrains fans
The machine-learning algorithm is largely successful at distinguishing white-painted carriages from snow, most weather conditions, and neatly stitches the resulting photos
SBB CFF FFS

> He was keen to use the Go programming language's computer vision and cross-compilation tools, and see how well it worked alongside older tech

creates an animation from a series of still shots. For his train version, Jonathan was keen to use the Go programming language's computer vision and cross-compilation tools, and see how well it worked alongside older tech such as SQLite and FTP.

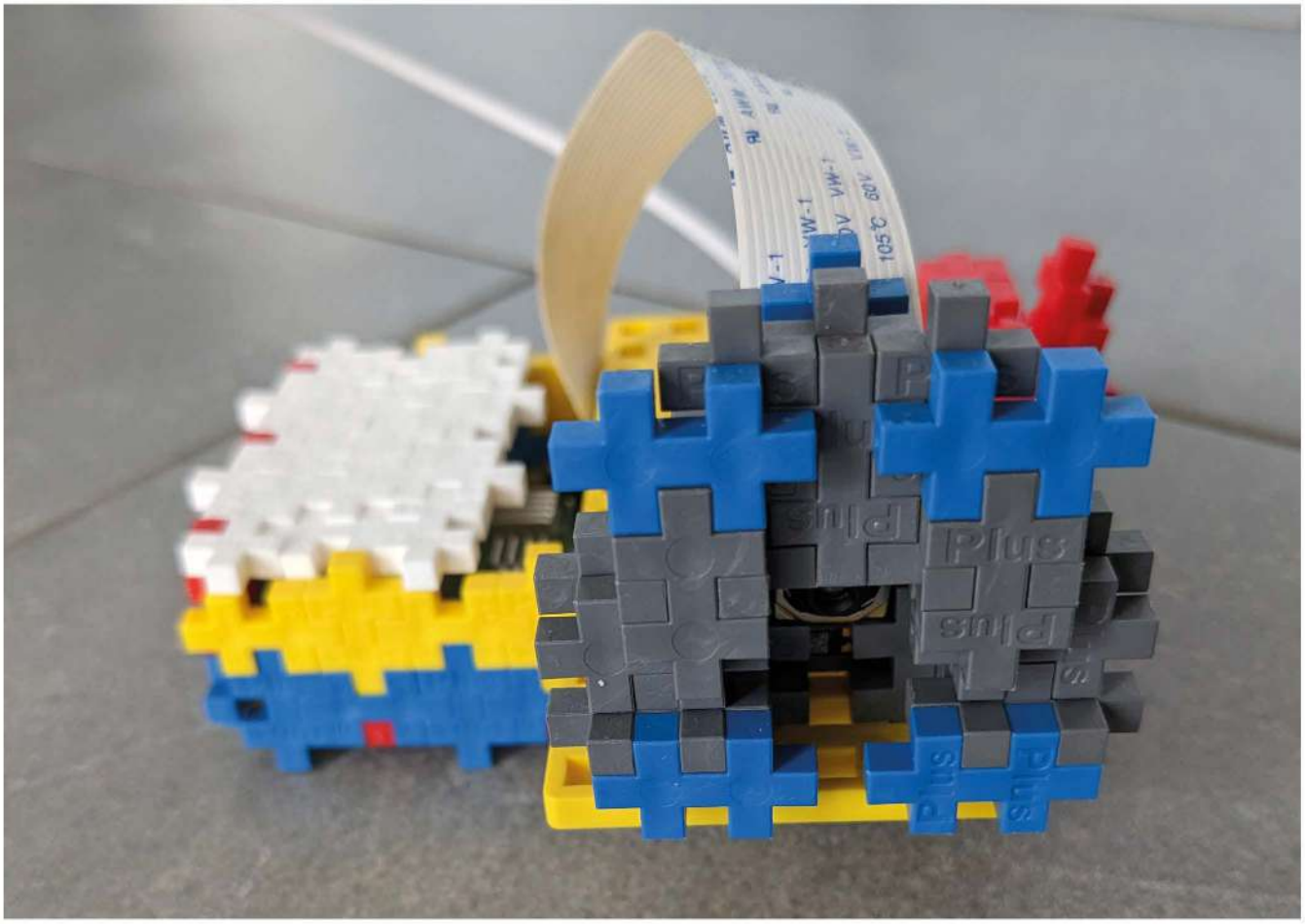
Jonathan chose Raspberry Pi 4 to run Trainbot, as "it offers all that is needed for such a project: quite a lot of compute power, cheap, good ergonomics and docs, and good software support." It is also powerful enough to handle the graphical computations required, while being less expensive than a board with a dedicated graphics processor. He notes that "the computer vision used in Trainbot is fairly naïve and simple. There is no camera calibration, image stabilisation, undistortion, perspective mapping, or 'real' object tracking."

Name that train!

With plenty of Raspberry Pi and Linux experience under his belt, Jonathan's challenges for the Trainbot included creating mounting plates and other small parts to fit inside the waterproof case he bought, so the AI camera setup could live outdoors on his balcony, as well as developing and testing the computer vision algorithm. He needed data with which to train the algorithm and spent "quite some time" recording video with his phone first, before "semi-manually chopping up the recordings and setting a test harness with them (**magpi.cc/autosetotestgo**)".

Quick FACTS

- Jonathan's day job involves work with drones
- Trainbot originally used a webcam housed in a kids' toy
- Better photos and a smartphone app broadened its appeal
- OnlyTrains fans can now save special shots as favourites
- You can see Jonathan's own favourites at **tinyurl.com/onlyfaves**



There was also a lot of "testing and tuning" involved in ensuring the computer vision setup was able to recognise most trains. "I soon figured out that patch-matching alone is not robust enough to estimate train velocity." Jonathan notes that this led to missed frames, or wrongly estimated ones, which led to chopped-up images. To address this, he added a step after the initial image gathering which applied an estimation and smoothing process using RANSAC and " a simple s = v0 + a*t model" to estimate start velocity and acceleration. RANSAC is a classical computer vision approach used to estimate a model when there are lots of outliers. He wrote much of the code himself, including instructions for patch-matching and the low-level computer vision, to keep things as light as possible.

With half of the coding and cross-compilation done in Go, Jonathan tried out Trainbot on his Raspberry Pi. At first, the project used a webcam but, as soon as Jonathan realised Raspberry Pi also makes camera modules, he switched over to HQ Camera Module 3, resulting in "much higher

▲ The mock-up model was housed in a toy casing and had a webcam for a camera

▶ Another of Jonathan's Raspberry Pi projects provides timetable information – something he's keen to integrate into Trainbot

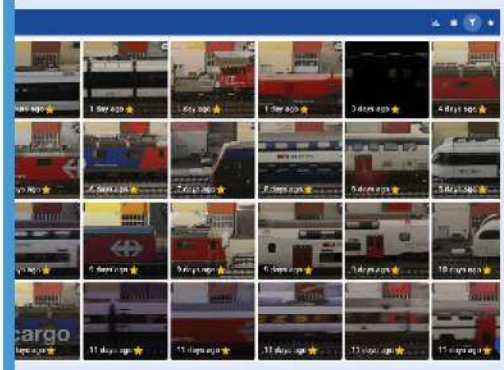
Raspberry Pi is protected inside a waterproof case

> I soon figured out that patch-matching alone is not robust enough to estimate train velocity

image quality (and very good software support via libcamera) than a cheap webcam, at a low price point." Next, he intends to develop the machine learning element to recognise and highlight 'atypical' trains, something that will be aided by him also adding details from the GTFS+/GTFS real-time timetables available in machine readable format from Swiss public transport companies.

Switching to Raspberry Pi HQ Camera Module 3 resulted in far superior photos, which Trainbot cleverly patches together

Set up a slit-scan



01 Use a tripod, or other fixing mechanism, to train your camera on a single spot. You will need plenty of footage samples of whatever you're photographing to train the software to auto-recognise their presence.

02 Use Jonathan's Go code to train the system and stitch together the images it recognises as belonging to the same vehicle or object: **magpi.cc/trainbot**.

03 Set up the camera, Raspberry Pi 4, and connectors in a sturdy weatherproof case where photos can be taken without the lens getting wet. Use FTP and SQLite to automatically upload photos to a web server for everyone to view.

Ohsillyscope

With Thomas McDonald's ingenious riff on the oscilloscope, live music performances don't just sound great, they look good too, as **David Crookes** discovers

MAKER

Thomas McDonald

Thomas graduated from UCCS with a major in Computer Science and a minor in Statistics. He's an outdoors enthusiast, dog dad, and music junkie.

magpi.cc/ohsillyscope

Thomas wanted a way to play his guitar and see a visual representation of the tunes on screen

Watch the official video for the Arctic Monkeys' Grammy-nominated single, *Do I Wanna Know?*, from 2013 and you'll see sound-waves vibrating in synchronisation with the percussion, lead guitar, and singer Alex Turner before turning into an animation.

It's a simple yet compelling effect and it certainly struck a chord with music fans, surpassing more than a billion views on YouTube in 2020. "My favourite part of going to concerts is always going to be the visuals and the music video definitely inspired me," says Thomas McDonald. It's led to the creation of an open-source version of an oscilloscope which displays live audio as a waveform. Thomas calls it the "Ohsillyscope".

The project allows Thomas to play musical instruments and see the tunes visualised as waves on a 64×64 LED matrix display. "After getting the LED matrix and playing around with it, I figured there would have to be a way that I could use it with my guitar and other musical instruments," he explains. "It's always really fun watching the live signal from synthesiser software, but I imagined I could make something more physical with the matrix."

Live jives

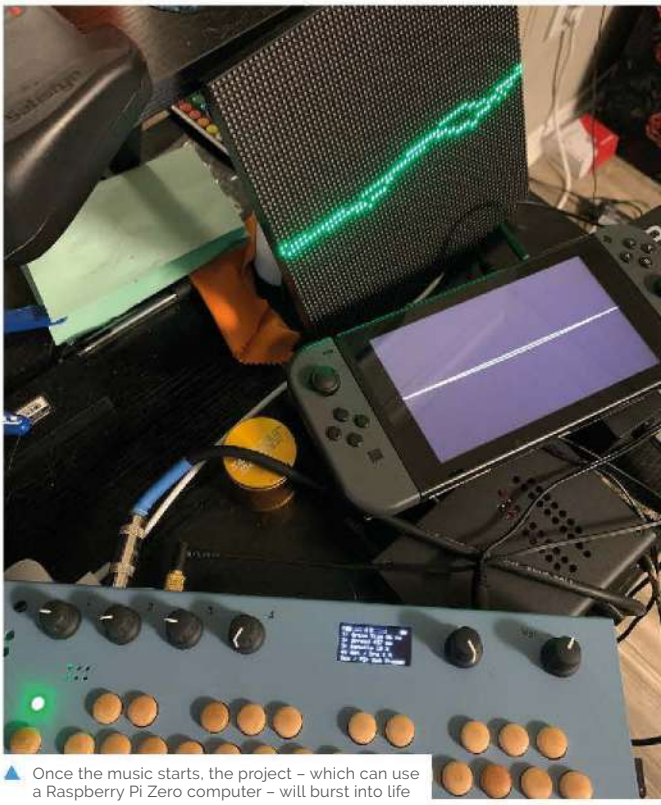
The idea, he explains, boiled down to registering a live audio signal with a Raspberry Pi computer. "Down the line, I want to make some more complicated visuals that are reactive to the audio signal coming in," he says.

It has involved connecting the matrix panel to an Adafruit RGB Matrix HAT add-on. "I have Raspberry Pi recognise the audio interface it's hooked up to, and I've used Linux's Advanced Linux Sound Architecture (ALSA) library to read the signal from the interface and place it into the C code running the LED matrix's library," Thomas says. "It's sampling at the refresh rate of the screen, which makes a really cool visual." Learn more at **alsa-project.org**.

The result is a project that can be taken on the road as a visual accompaniment to musical performances. "At first, I was thinking that I could have it work through MIDI [a protocol which lets musical instruments, computers, and other hardware communicate with each other], but the idea of needing to be hooked up to the software Ableton Live killed the motivation to write an entire library to do so. I wanted something that didn't require being by a computer to represent music."

Looking good

Thomas loves using his creation. "It's small and portable – something you just plug in knowing that



Once the music starts, the project – which can use a Raspberry Pi Zero computer – will burst into life

The Adafruit RGB Matrix HAT means setting up the Ohsillyscope is rather straightforward

The Ohsillyscope reads live audio from ALSA sources

The 64×64 LED matrix displays a waveform image in real-time and it can do so in various colours

Quick FACTS

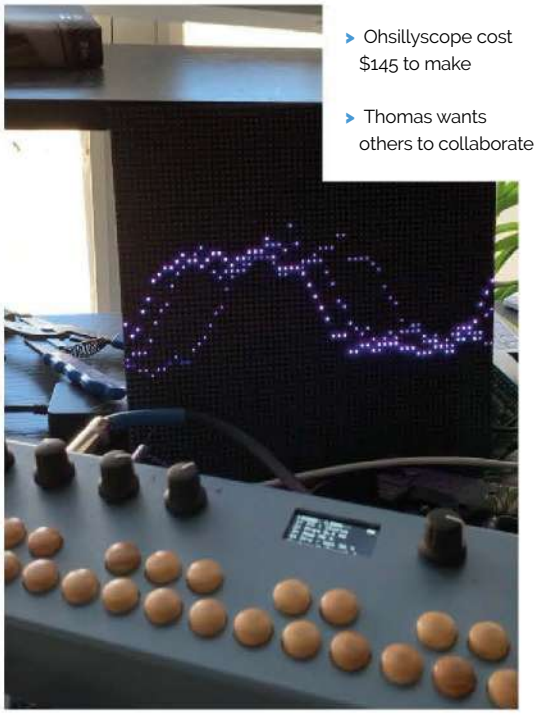
- The device visually represents music
- It displays it on an LED matrix
- ALSA is used to interpret an audio signal
- Ohsillyscope cost $145 to make
- Thomas wants others to collaborate

everything is set up," he says. "The LED matrix HAT was made for Raspberry Pi, so it was a no-brainer. It also helped that I had a couple of Raspberry Pi Zero computers laying around begging to be used."

Currently, he's hooked the Ohsillyscope to the PA system in his band practice room. "It picks up the bass, guitars and drums, making an interactive

> It's sampling at the refresh rate of the screen

show for anyone watching," he says. "The people we've shown the Ohsillyscope to so far seem to love it and once we start gigging more, it's sure to get the band more attention." We're definitely sure that it's going to look good on the dance floor.



You can create your own version of the Ohsillyscope since the code is open-source

Olga The Fortune Teller

Admiring glances across an office led to a creative partnership involving several Raspberry Pi models and some charming AI, learns **Rosie Hattersley**

MAKER

Kakapo Labs: India, Will Lahr; AlterD: Will Lucking, Claire Richards

AlterD creates retail displays for clients and shares office space with interactive design agency Kakapo Labs, which routinely uses Raspberry Pi in its creations.

magpi.cc/kakapolabs

Last issue, we brought you details of the Love Machine ChatGPT project which dished out compliments and sometimes sweets based on cutesy WhatsApp exchanges. Raspberry Pi staffers and *The MagPi* magazine readers weren't the only ones to fall for its AI charms. A company that shares office space with Love Machine makers Kakapo Labs were equally entranced, and soon made a move, proposing a coupling up and a glow-up! The result: Olga The Fortune Teller, a shiny new ChatGPT model designed to appeal to the Insta-savvy influencer keen to check out both the latest skincare options and receive reassuring messages about their future fame and follower count.

Altered images

Olga The Fortune Teller was designed in partnership with AlterD, a company that designs displays for retailers. Both companies are based at Hone (**hone.london**), a creative space in East London. The team from AlterD was on the look out "for something visually attention-grabbing", saw Kakapo's Love Machine, got talking to its makers, and things went from there.

With a retail display trade show coming up imminently, AlterD chose a cardboard-based design to show off the material's possibilities. "It was important that it didn't look too specifically like any particular brand, so they went for a fun, quirky, dream-like look. They wanted to add an interactive element to encourage people to interact with the stand and start conversations," Kakapo Labs founder Will Lahr explains.

As with Love Machine, mobile connectivity was vital, so they chose a Pico W: "simple, low-cost and small but, at the same time, [it] can run a full

Olga's beauty secrets involve lashings of perspex, cushioning cardboard, and colour-changing mood lighting, plus some Pico W-controlled eyebrow raising

Once you've checked your visage in the helpfully supplied make-up mirror, scan the QR code below it to engage Olga in a WhatsApp chat

Olga gets ChatGPT to foretell your fortune, sending it to a cloud server for Pico W to download and deliver as printed ticket

Quick **FACTS**

- Olga cost around £160 to build
- It was originally going to be a visual guessing game
- Instead, she became a 'director of divination'
- She features ChatGPT's new chat Completion model
- It can tell the fortunes of multiple WhatsApp users

▲ Olga's gorgeous face and expressive eyebrows were laser-cut

wireless stack which is easy to use." With MicroPython on the Pico W, things work painlessly and the development time is short. It's used here to communicate with a Zebra thermal printer via an RS232 connection and prompt it to print out a stand visitor's fortune ticket. ChatGPT is instructed to: 'Pretend you are a fortune teller and tell the person you are talking to – who is called $NAME – their fortune in a poem of three or four short lines. Make sure their fortune is vague but positive! Include a lucky item (retail related).'

A concurrent request is made to DALL-E (an image generator made by the same people as ChatGPT for a black and white cartoon image of a fortune teller. Fortune text and graphic are imported and encoded as an image that can be recognised by ZPL, the language used by the printer. Laggy performance processing and outputting the images led the Kakapo and AlterD team to switch from the unreliable existing WebSocket delivery setup. The "much better" system they implemented saves the ticket image online and just uses WebSocket to send Pico W the URL in order to download and print it.

Fortune favours

Olga herself is, of course, a large part of the appeal. She has a face cut from a sheet of perspex and some extremely expressive eyebrows thanks to a Pico W, micro servos, and some black thread that help them wiggle. She also imparts her wisdom via a fancy new means of communicating (due to changes to ChatGPT that OpenAI recently implemented), but it's still based on WhatsApp and Twilio, like Olga's sibling, the Love Machine (see *The MagPi* 130, **magpi.cc/130**).

The brief from AlterD was for a playful interactive display showing off the possibilities for retail settings. However, Will points out that creating something very quickly and efficiently using several Pico Ws (which, being connected

▶ Colour-changing LEDs make the booth design more eye-catching and hide Olga's cardboard outline

> Olga herself is, of course, a large part of the appeal. She has a face cut from a sheet of perspex and some extremely expressive eyebrows

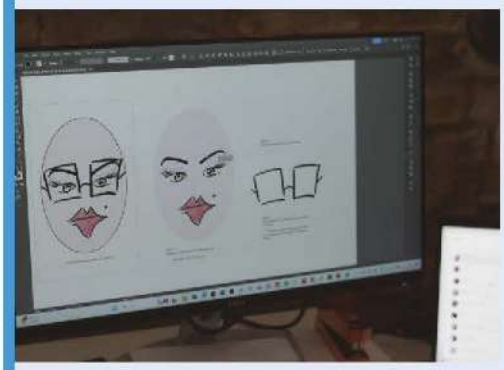
wirelessly, avoids the need for long cable runs), making use of simple web services has all sorts of potential. "We've been exploring more possibilities for interaction through a shop window – using a phone as a device to let someone outside the window play with something inside the window," he says, and is up for collaborating with other partners for more fun, playful projects.

"Olga may not be able to predict the lottery or tell us when the cost of living crisis will end, but what she can do is offer a glimpse into the future of interactive retail experiences."

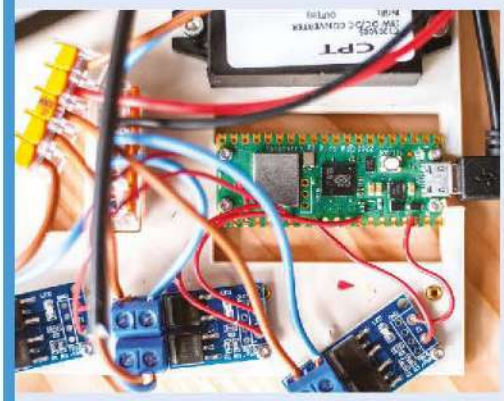
Olga can hypnotise even the most casual passer-by into having their fortune told...

After a brief WhatsApp chat, the booth prints out a pronouncement

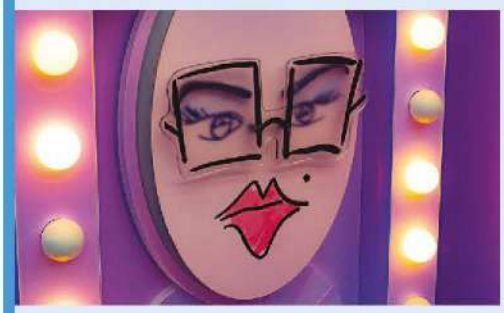
Characterful interaction



01 Olga's look is very much down to you. Create the fortune teller's face with or without a computer using items such as cardboard and Perspex, adding micro servos and Pico W if you'd like to include changing facial expressions.



02 Pico W controls the LED lights around Olga's make-up mirror, as well as the servo motors creating her facial expressions, while another connects to the Zebra printer to produce the all-important fortune readings.



03 Use the instruction in the main article to get ChatGPT to tell a fortune. Connect everything to a web server and thermal printer to print them out on demand.

Chonky Pocket

Dan Norris's project looks like a video game handheld console, but it's a computer that can be operated with one hand. **David Crookes** takes a look

MAKER

Dan Norris

Dan is a DevOps and infrastructure engineer in Northern California who enjoys hiking and building stuff

magpi.cc/chonkypocket

On-screen keyboards work well, but a physical set of keys remains preferable, certainly if you're going to be entering a lot of text. With this in mind, Dan Norris began to work on his Chonky Pocket computer. He believed that, in order to create a device comfortable to use, jamming in a full-size Qwerty keyboard wasn't going to work.

Instead, he has limited his pocket-sized computer to just ten keys and, as well as being able to type straightforward documents with it, it's geared up specifically to allow users to code in Python. What's more, it's wrapped in a beautiful enclosure that makes the Chonky resemble a Nintendo Game Boy handheld. It really is an object to desire.

"I wanted a pocket-sized Linux computer with a physical keyboard and I sort of hacked something together last year that wasn't very nice," Dan explains. "I decided to have another go at it this spring when I got a pre-release Amp Ripper 4000 PSU to try out." His goal was to incorporate a keyboard, scroll wheel, and touchscreen, and include a laptop-style battery that would report its state of charge to the OS.

This is an early draft build of the project with a vertical screen and a GPIO keyboard

Key points

At the Chonky Pocket's heart is a Raspberry Pi 4 computer. "I usually use Ubuntu and it's great that Raspberry Pi is officially supported," Dan says. It's hooked up to a 5-inch HDMI touchscreen with a speaker and, of course, a handful of keys. They're 'chorded' which means: several keys need to be pressed together to enter characters, words, phrases, or commands.

"The keyboard layout is mostly just the eight-key ARTSEYIO (**artsey.io**) chording layout with two extra keys mapped to ')' and '(' and chorded ':' – keys that normally require a layer shift with the ARTSEYIO layout." This enables Python coding. "It was a bonus that the extra keys made the keyboard physically symmetrical, so I could change the layout for either hand," Dan adds.

Of course, it's going to take some getting used to. "I can type about 20 words per minute (wpm) with it when I practise a bit, but there are folks on the ARTSEYIO Discord who can do more than 40 wpm," Dan continues. "If I wanted to type faster with one hand, I'd be inclined to use half of a Corne split keyboard (CRKBD) which has more thumb buttons for layer shifting and more direct access to symbols, numbers, and CTRL/SHIFT/ALT."

Powering on

The build posed some challenges. "Modifying the battery kernel module for this was interesting," Dan says. "I haven't touched any C code since college and the window manager would crash when I loaded the module in Raspberry Pi OS. I'm just lucky that it worked in Ubuntu without having to do much troubleshooting."

Power was hard to get right too. "Even using the wrong connector between a battery and the boost/charge board can cause voltage warnings

The Game Boy-style case needs a minimum thickness of 32 mm to fit the display and Raspberry Pi 4

The mechanical keyboard uses low-profile Choc switches with keycaps. The keyboard uses chording and has mouse emulation

Dan says a lot of apps work nicely with a mouse scroll wheel ,even if you don't have a mouse

Quick FACTS

- This is Dan's tenth cyberdeck-related project
- It uses a chording keyboard
- The Game Boy-style case helped simplify the design
- All the STL files are on Dan's GitHub page
- There's a gyro and accelerometer inside too

The Raspberry Pi computer ports can be accessed from the back. There's also a kickstand, power switch, and accessory mount

and rebooting with Raspberry Pi 4," he explains. And he has rewritten the GPIO keyboard firmware a couple of times. "It was an opportunity to learn about the current state of GPIO programming on Raspberry Pi with 'lgpio' and figure out how to use 'uinput'."

> I usually use Ubuntu and it's great that Raspberry Pi is officially supported

Dan hopes many people will enjoy building their own version and already plans for a build that includes a second HDMI port, external pins, a display that turns off when idle, and more. "I've gotten lots of supportive comments on Reddit for this build and some nice media coverage. It's been a lot of fun," he says.

The batteries will last for up to a day: "They're heavy and bulky," Dan says. "But it's cool not having to drag around a charger"

GPS Pothole Tracker

If problematic potholes drive you round the bend, this practical device might improve their chances of being repaired. **Nicola King** checks her tyres

MAKER

ScienceDude1990

ScienceDude1990 is a maker in Canada, and an electrical engineer by training. He enjoys making off-beat projects, and writing well-commented and clear code.

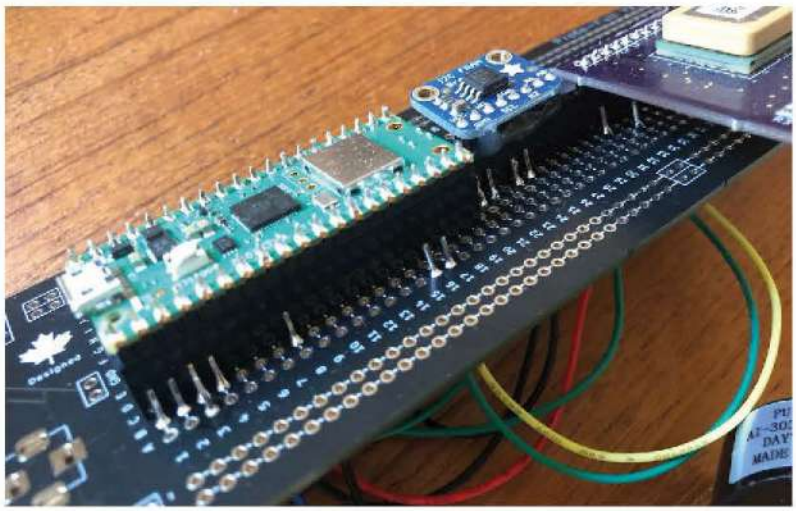
magpi.cc/potholetracker

Potholes are becoming an increasing problem the world over – on regular trips, you get to know the spots on the road to avoid and, when you find a new hazardous hole on your highway, you and your suspension certainly feel the sometimes damaging effects. So, what if you had a way of recording the exact positions of these troublesome fissures and fractures, so that you could then alert the relevant authorities to fix them?

Finger on the button

Well, it's a nuisance that ScienceDude1990 decided to tackle, following one particularly hard and pothole-inducing winter in his native Canada. He was determined to try and deal with the headache of his local potholes for the greater good: "I think it is my civic duty to help out. Potholes can be annoying for cars, and very dangerous for cyclists."

After gathering a few components, a little breadboarding, coding, and soldering ensued, and the result of his endeavours is the GPS Pothole Tracker, the key elements of which are a Raspberry Pi Pico W, an Adafruit LED arcade button, a buzzer, a u-blox antenna module (for GPS), and a FRAM (ferroelectric RAM) breakout for data storage.



To make the project parts stable but reusable, he soldered rows of headers to plug the boards into

For portable power, the GPS Pothole Tracker uses a USB power bank. Out in the field, when the user is near a pothole, they press the button. If the GPS is locked, the tracker records the button press in the FRAM. "The Pico W takes in the input from the GPS, gaming button, and then stores the GPS locations, and gives output to the user in the form of the light of the button, and the buzzer," explains ScienceDude1990. When the user gets home, they can then open a text file and send any pertinent pothole information onto whoever needs to have it.

ScienceDude1990 says that Raspberry Pi Pico W was easy to use with MicroPython and Thonny when coding for this device. "The development environment is very lean, fast to learn, and perfect for these kinds of projects." The most challenging aspect was learning how to use the 'os' commands in MicroPython to work with the FRAM board.

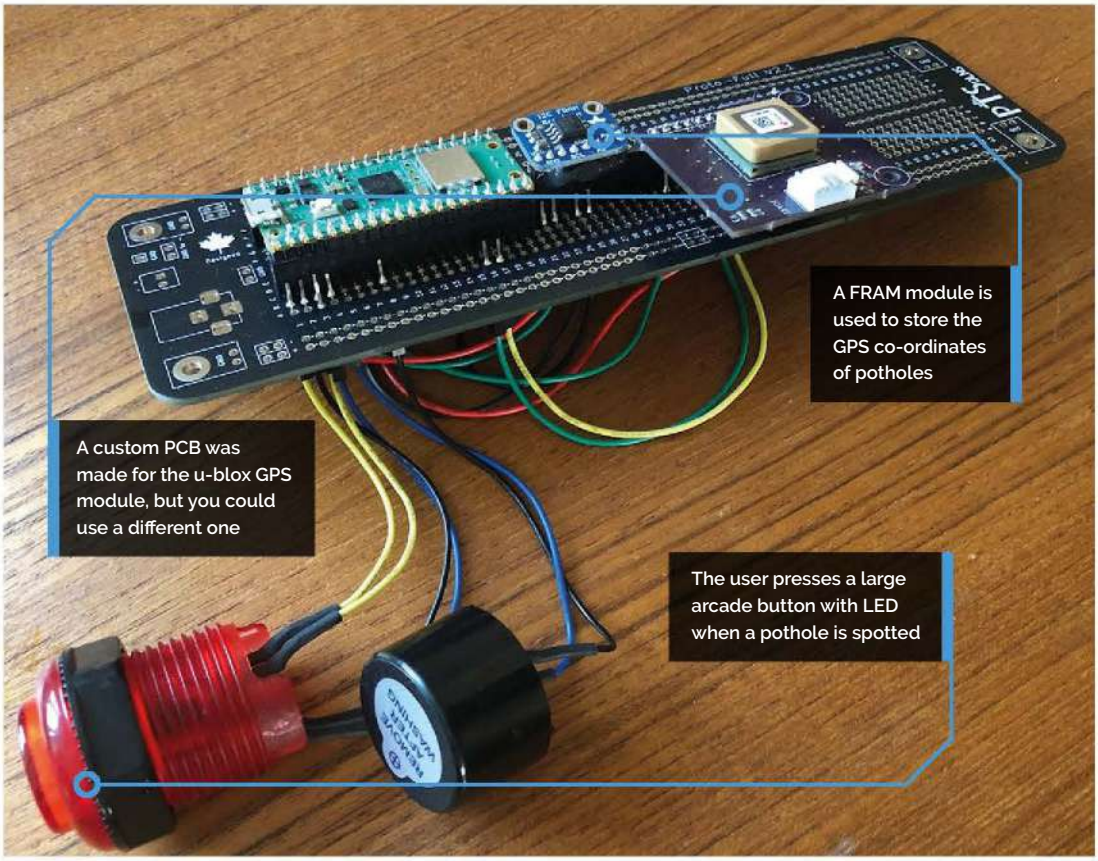
> "Potholes can be annoying for cars, and very dangerous for cyclists"

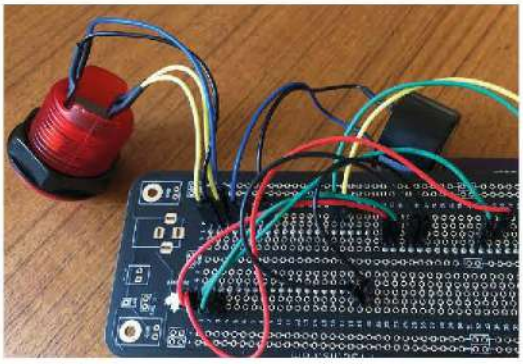
He describes using the FRAM as a "side quest", as he was keen to see how he could work with the FRAM chip and the Pico W, and says he was "pleasantly surprised" at how easy it was to format, mount, and use.

Mission accomplished

The good news is that ScienceDude1990 has actually used his device on his local roads, alerted his obliging local authorities to the presence of pesky potholes, and got the job done! "I used it to log the potholes in the spring, and then submitted them to my city's service request system. The potholes were fixed in a timely manner. Joy!"

All that's needed now is a case of some kind to protect the various components and make it a little more aesthetically pleasing, as currently the device is very much in its raw form. "Definitely going to

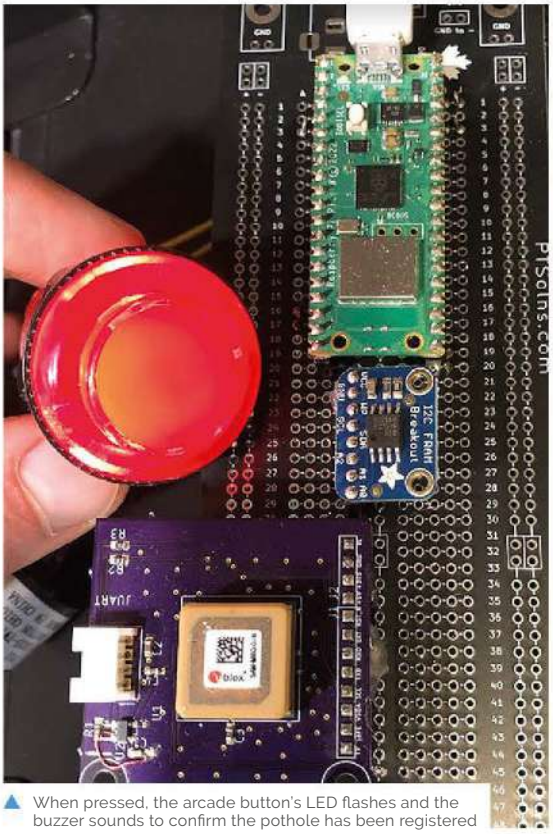

A custom PCB was made for the u-blox GPS module, but you could use a different one
A FRAM module is used to store the GPS co-ordinates of potholes
The user presses a large arcade button with LED when a pothole is spotted



▲ The wiring is all on the rear of the prototyping board, including connections to the button and buzzer

make a nice case for it," he says. "[I'll] probably laser-cut something out of clear acrylic at the library makerspace."

It's a relatively simple make, but ScienceDude1990 suggests a little bit of soldering practice might be useful before a complete newbie attempts to make one. "I would suggest breadboarding with push-in wires first. Then practise soldering, or use some kind of quick-connector scheme. His informative Hackaday project page (**magpi.cc/potholetracker**) gives links to the GitHub code he wrote, and includes a Pico pinout connection diagram.



▲ When pressed, the arcade button's LED flashes and the buzzer sounds to confirm the pothole has been registered

Warning!
Drive safely

If you make this project, ensure safety comes first behind the wheel. As ScienceDude1990 notes: "Always remember to respect your distracted driving laws, so maybe get a co-pilot to help?"

magpi.cc/drivesafely

Quick **FACTS**

- The total cost of the make was just under Can$95
- The largest line item was the GPS module – the part cost around $50
- ScienceDude1990 spent around two weeks' worth of free time on the project
- His project code is available on GitHub: **magpi.cc/potholetrackergh**
- He now plans to make an oscilloscope with Pico's PIO functionality

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W

+

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 23% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- Free delivery to your door
- £10 (UK only)

Subscribe for 6 Months

- Free Pico W
- 6 issues of The MagPi
- Free delivery to your door

£30 (UK) $43 (USA)
€43 (EU) £45 (Rest of World)

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a 6-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

WORTH $6

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Python Robots
Build dynamic bots with arms and legs
CREATE A RASPBERRY PI WILDLIFE CAMERA

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

SUCCESS STORY magpi.cc/success

Sisyphus

Powered by Raspberry Pi, this mesmerising artwork is an international hit. By **Rosie Hattersley**

Artist Bruce Shapiro creates mesmeric kinetic art that constantly undulates, fascinating the viewer. He grew up "enchanted by music, electronics, and making things" and took tentative steps towards his own creations using motors from unwanted floppy disk drives and a 286 computer. He quickly realised that discarded industrial automation parts could be picked up for peanuts via surplus outlets, and that CNC machines and motors were effective for motion-control tasks.

Bruce's knack for creating kinetic art as a means of self-expression took off and, after originally working in medicine, his experiments in electronics and computing control led to a new career. He became a solo kinetic artist and science educator, using motion control technology to create large scale installations for museums. (His eye-catching artworks continue to be displayed in museums and art galleries in Australia, Germany, Switzerland, and USA.)

In 2015, Bruce created a new kinetic art table, Sisyphus, again using CNC and polar stepper motors. This time, the precision CNC machine that Bruce used to create cutouts for his kinetic sculptures became a piece of art in its own right. The Sisyphus Table took inspiration from the legend of Sisyphus constantly pushing a boulder

up a mountain: in Bruce's work, a ball bearing in place of a rock creates a constantly shifting pattern on a layer of sand. A stepper-controlled magnet underneath the sand table directs the ball bearing's motion. The robot controlling everything, the Sisbot, used the same essential design Bruce had invented for his first motion control device back in 1990.

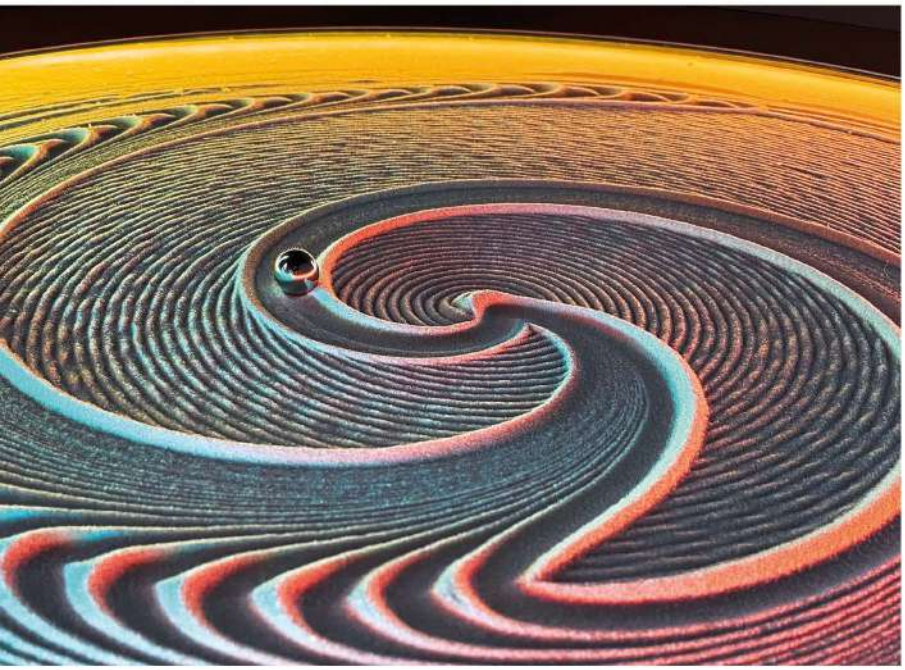
THE CHALLENGE

There was plenty of interest in the design from the art world and science museums, so Bruce founded Sisyphus Industries and set about designing a sand table with dual function as furniture and a playable piece of art. Crowdfunding seemed an obvious route. The choice of operating software and hardware to drive it was less so – until Bruce spoke to those he knew and trusted. "It was clear that smaller, less expensive control hardware would be needed."

THE SOLUTION

Hardware costs, reliability, and community support for the technology driving the kinetic table led Bruce to consult his software engineer son, the embedded software engineer he was working with, and the younger members of the nearby makerspace in Minneapolis. The vote for Raspberry Pi was unanimous.

It proved to be a good fit. The sand-plotter



mechanism, the Sisbot, needed to connect to a custom two-motor driver board via USB serial port. Other output includes control of an addressable RGBW LED strip. Inputs include two Hall effect home sensors, and an analogue signal from a phototransistor used to detect ambient light levels (for auto-dimming the LEDs). Raspberry Pi could do all of this. Both commercial and large-scale installation versions of Sisyphus are controlled by Wi-Fi-connected Raspberry Pi computers running JavaScript on Node.js.

> Our experience using thousands of Raspberry Pis has been relatively seamless

WHY RASPBERRY PI?

In commercialising something that was previously simply an art installation, issues of customer support and maintenance became important. In addition, some "painful" dead ends lead the change-averse artist to look for a new means of powering and controlling his Sisbot. "I don't like change when it comes to something that works," Bruce admits. "Through many trials, and sometimes painful dead ends, I've learned that community matters."

It was the Raspberry Pi community that convinced Bruce to switch over to a low-cost microcomputer. "This was even more important than form factor and low cost," says Bruce.

"The choice to use Raspberry Pi in the new home versions of Sisyphus was driven by all the thirty-somethings that I listen to," says Bruce. The community informed him that JavaScript is essential and Node.js runs well on a Raspberry Pi.

THE RESULTS

"As a musical instrument plays songs, Sisyphus plays paths," the 2016 Kickstarter pitch explained. The aim: to "get Sisyphus into peoples' homes for them to enjoy as both furniture and art, but also, to inspire a community of composers to write 'music' for it." The Sisyphus Sand Table became – and remains – the most funded art project in Kickstarter's history, Bruce tells us, proudly.

Five years after launching Sisyphus Industries, the company now has 15 full-time employees at its Minnesota headquarters, and recently shipped its ten-thousandth Sisyphus Table, with customers across the globe and recent installations in Dubai and Oslo.

"Looking back over this time, our experience using thousands of Raspberry Pis, including 3B, 3B+, 3A+, and now 4, has been relatively seamless," says Bruce. "Our software developers have been able to rapidly deploy our firmware across different Pi versions. The hardware has proved highly reliable, both in the production facility and in the field. And the original reason we chose the Raspberry Pi – its vast community – has proven to be invaluable in recent experiments using kinetic art within STEM education."

Images by Scott Streble used with permission. Other images are by Sisyphus Industries

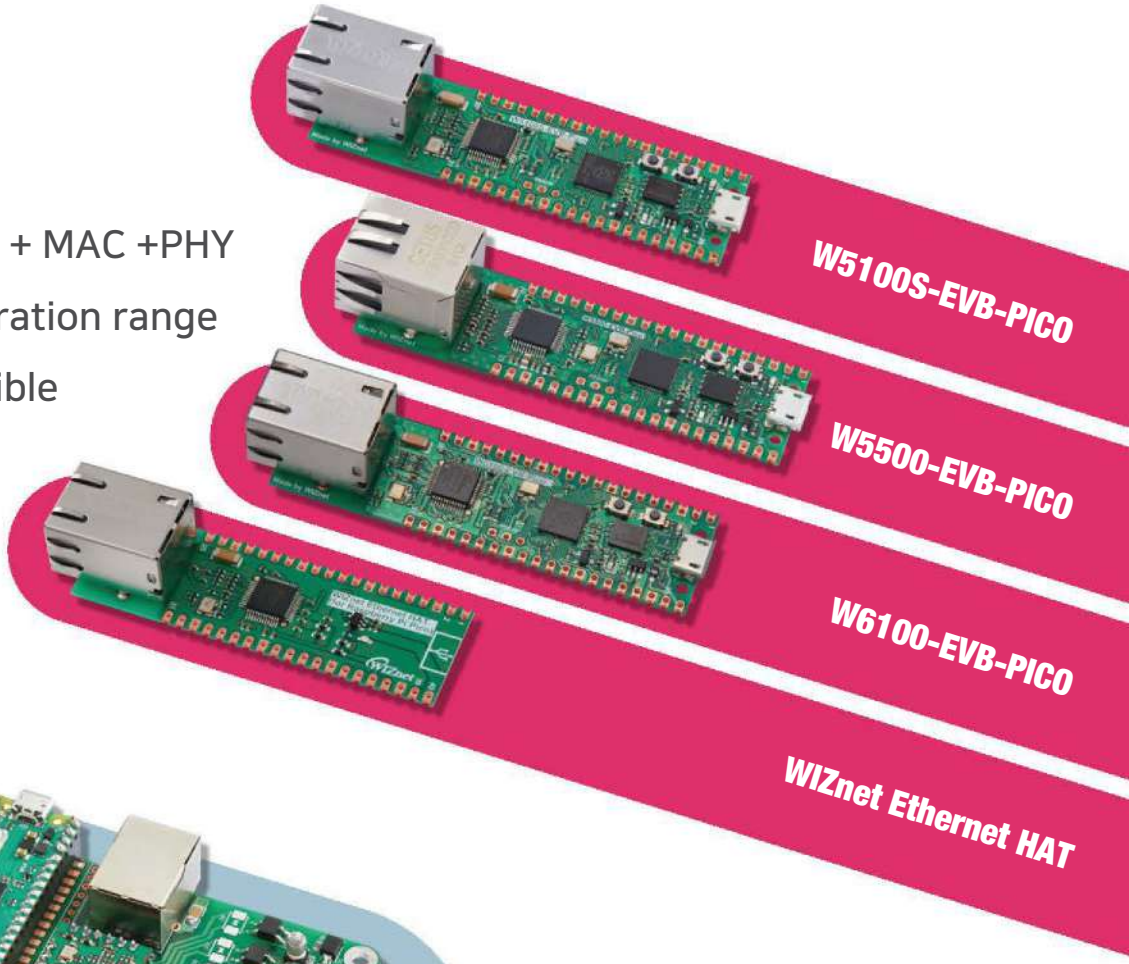
WIZnet

Meet the
WIZnet Pico family

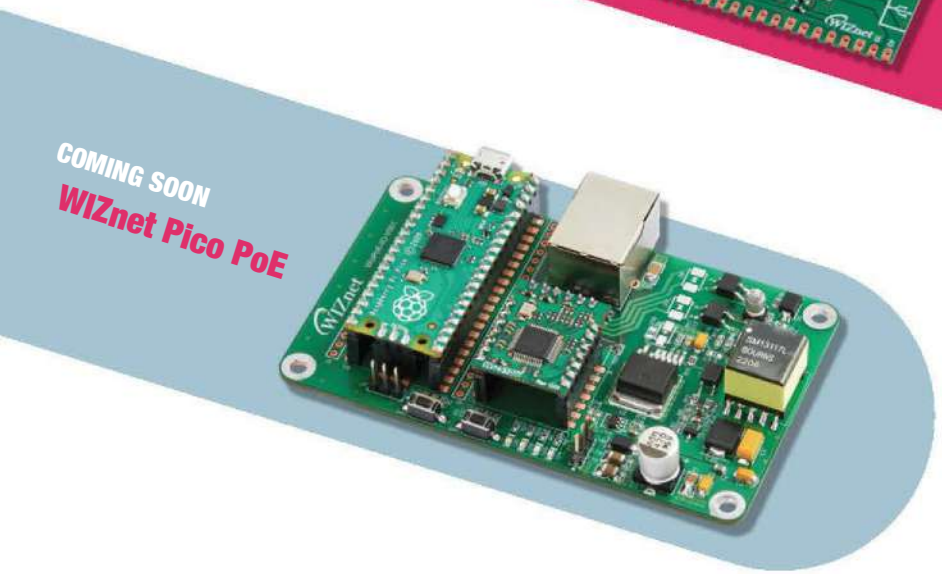
Experience the Power of Connectivity with WIZnet

Powered by Raspberry Pi

- 3-in-1: TCP/IP Stack + MAC +PHY
- Industrial grade operation range
- RPi Pico pin-compatible



COMING SOON
WIZnet Pico PoE



- IEEE802.3af compliant
- Mode A(Endspan), Mode B(Midspan)
- Circuit Protection (OV,OC)

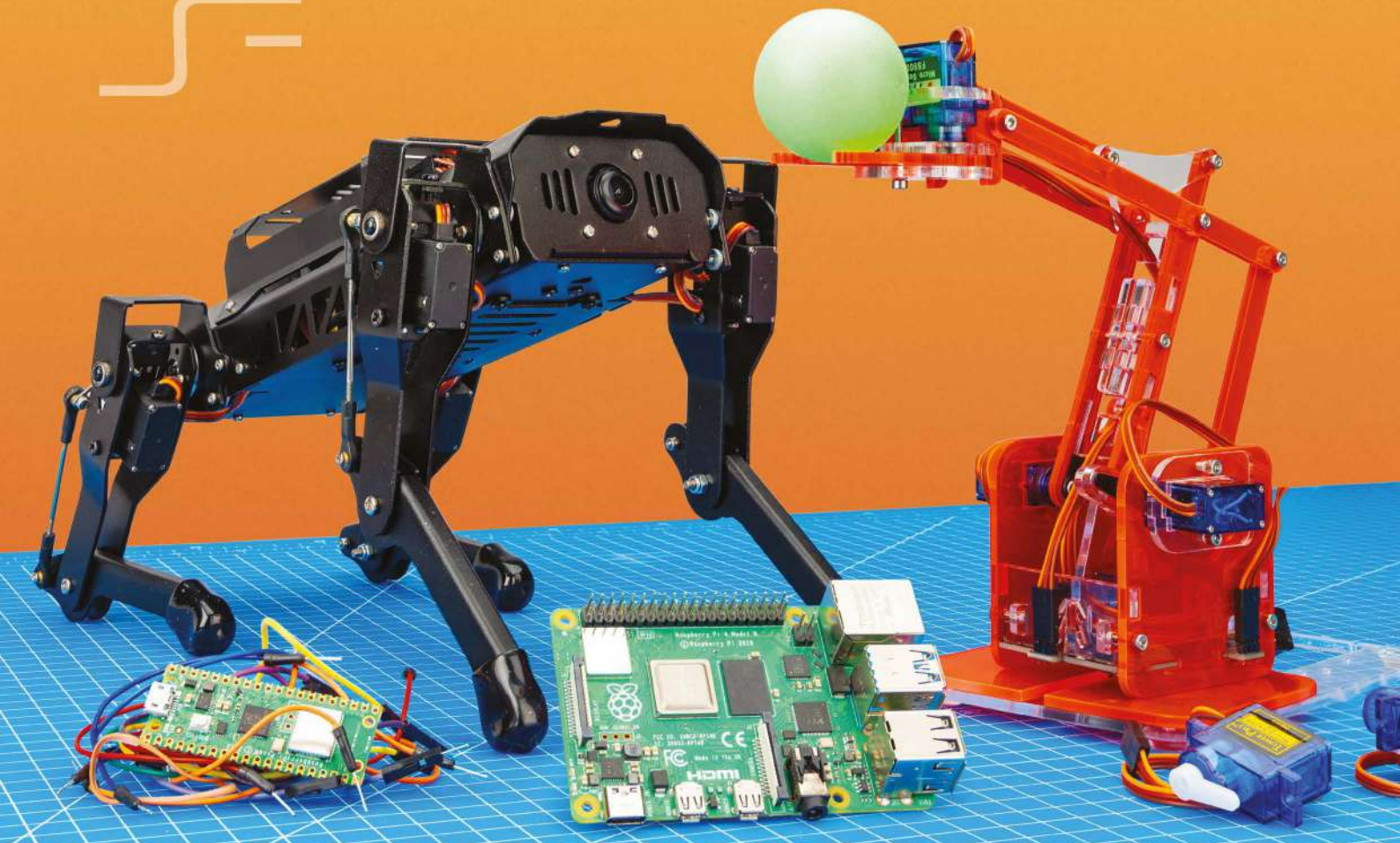
Reseller suggestions and inquiries direct@wiznet.io

Available at WIZnet online store eshop.wiznet.io

Python Robots

Craft, control, and program Python-powered robot arms and legs

By Danny Staple and Lucy Hattersley



Robots become infinitely cooler when arms and legs power them. This feature has them all, including walking dogs, humanoid replicants, and grasping robot arms.

These robots all share one thing in common: servo motors, also known as 'servos'. This type of motor rotates back and forth, unlike the DC motors of wheeled robots that spin endlessly around.

> This feature will equip you with a real-world, hands-on understanding of articulated robotics

Dive into an enriching journey of Python programming and robotics with this feature. We will explain how servo motors work, how you can control them in Raspberry Pi and look at some of the fantastic robotic kits on the market.

Whether you're a hobbyist or an aspiring robotics expert, this guide will equip you with a real-world, hands-on understanding of articulated robotics. So it's time to gear up, assemble your tools, and embark on a thrilling robotics adventure!

TYPES OF SERVO

Servo motors are a type of motor that rotates back and forth with precision. They are used in robots that require precise control, such as robot arms and limbed dogs and walking bots. The more expensive options include a sensor of some kind to measure position. Here are some types of servo to look out for...

SG (Servo Gear)

Servo gear units like the SG90 and newer SG92 (£4/$4) are inexpensive and light with precise movement control. The SG92R can rotate approximately 180 degrees (90 in each direction). You'll find them in entry-level robotics projects where cost and weight is a concern.

magpi.cc/sg92

MG (Metal Gear)

The next step up in servos is to switch out the plastic gears for metal. Servos such as this MG92 (£7/$7) offer more durability and strength than the entry-level SG models, and are better where the servo is going to lift weights or move loads. They are a bit heavier and noisier than their plastic counterparts.

magpi.cc/mg90

Brushless

In a typical motor, brushes are used to switch the polarity of the magnetic field, which is necessary for the motor to continue spinning. However, brushes are subject to wear and can create electrical noise, limiting the lifespan and performance of the motor. Brushless motors last longer, use less energy, create less heat and make less noise while being faster with further range. They cost a lot more, though.

magpi.cc/brushlessservo

ROBOT KITS

MeArm Maker Kit

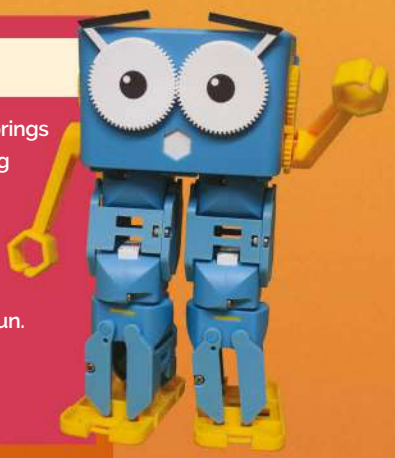
This is the kit that we are using to explain servo control for the main part of this feature. MeArm Maker (£39/$50) is a very low-cost robot arm that can be assembled with just a hex key "and a dash of enthusiasm". The kit includes laser-cut acrylic parts and four servo motors. It's basic, but that simplicity makes this a great place to start before moving on to more complex walking robots.

magpi.cc/mearmmaker

Marty v2

Marty the Robot V2 (£415/$449) is a charming two-legged robot that brings coding to life. Developed by Robotical, it features lots of accompanying educational projects and this friendly robot is controlled using Scratch or Python. Robotical claims that it is suitable for children aged seven and up to university level. Marty is controlled using coloured cards, an Android or iPhone app, or Raspberry Pi. Despite its cute exterior, Marty is packed with potential, making it a great tool for learning and fun.

magpi.cc/marty



Petoi Bittle X

We tested Petoi Bittle (£204/$256) and found it to be a cute, palm-sized robot dog that is a great tool for promoting STEM learning. A new Bittle X model is in for testing and now offers voice interaction. It is a quad robot with four controllable legs and its price is a more affordable alternative to high-end robot pets. It comes with the NyBoard 1 controller board and ten servo motors. It's a delightful way to own a robot dog and there are a lot of coding projects to be found here.

magpi.cc/petoibittle

Tony Pi

Hiwonder has impressed us with its wide range of high-quality articulated robots. TonyPi (£453/$570) is a humanoid kit powered by Raspberry Pi 4B 4GB. It's not cheap, but the construction is superb and it features high-voltage bus servos. This advanced robot is designed to provide a hands-on experience in AI and robotics. TonyPi's capabilities include image recognition and tracking, making it a versatile tool for learning and exploration. Hiwonder provides an app interface, and you can dive into the kit via SSH and code it directly, so it's good for beginners and experts alike.

magpi.cc/tonypi

SpiderPi

The Hiwonder SpiderPi (£477/$600) is a substantial robot, standing nearly 30 cm off the ground and spanning nearly 60 cm when its legs are fully extended. Constructed with 18 servos similar to its Hiwonder sibling, the TonyPi, the SpiderPi is controlled by a 4GB Raspberry Pi 4. This hexapod robot features an HD camera that can rotate through 140 degrees, enabling it to detect faces and signal recognition by waving a leg.

SpiderPi performs various movements including scuttling forward, back, and sideways, as well as twisting and raising its front legs. Its camera and movement speed can be adjusted via a sliding bar in the app. The robot can also follow an undulating line on the floor, distinguish between different-coloured objects, and follow commands issued via a Python script tied to a QR code. It scared the bejesus out of our cat.

magpi.cc/spiderpi

Mini Pupper

MangDang's Mini Pupper (from £357/$449) is an open-source ROS robot dog kit designed to make robotics more accessible. MangDang claims Mini Pupper to be an excellent hands-on tool for STEM and K12 education. With this kit, learning robotics becomes a fun and engaging journey.

magpi.cc/minipupper

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
SUPER SIMPLE ROBOTICS
YES! YOU CAN BUILD THAT ROBOT
BACK TO SCHOOL WITH RASPBERRY PI

SUPER SIMPLE ROBOTICS

We've got a guide for newcomers to robotics in issue 120 of *The MagPi*. Unlike the articulated wonders featured in this tutorial, our earlier feature shows you how to create a wheeled robot using a few low-cost parts and control it with code. Wheeled robots are much simpler, but less fun.

magpi.cc/120

Control a robot arm with Python

Build and control a robot arm, while learning about how to remotely control servo motors on Raspberry Pi Pico W

By **Danny Staple**

DANNY STAPLE

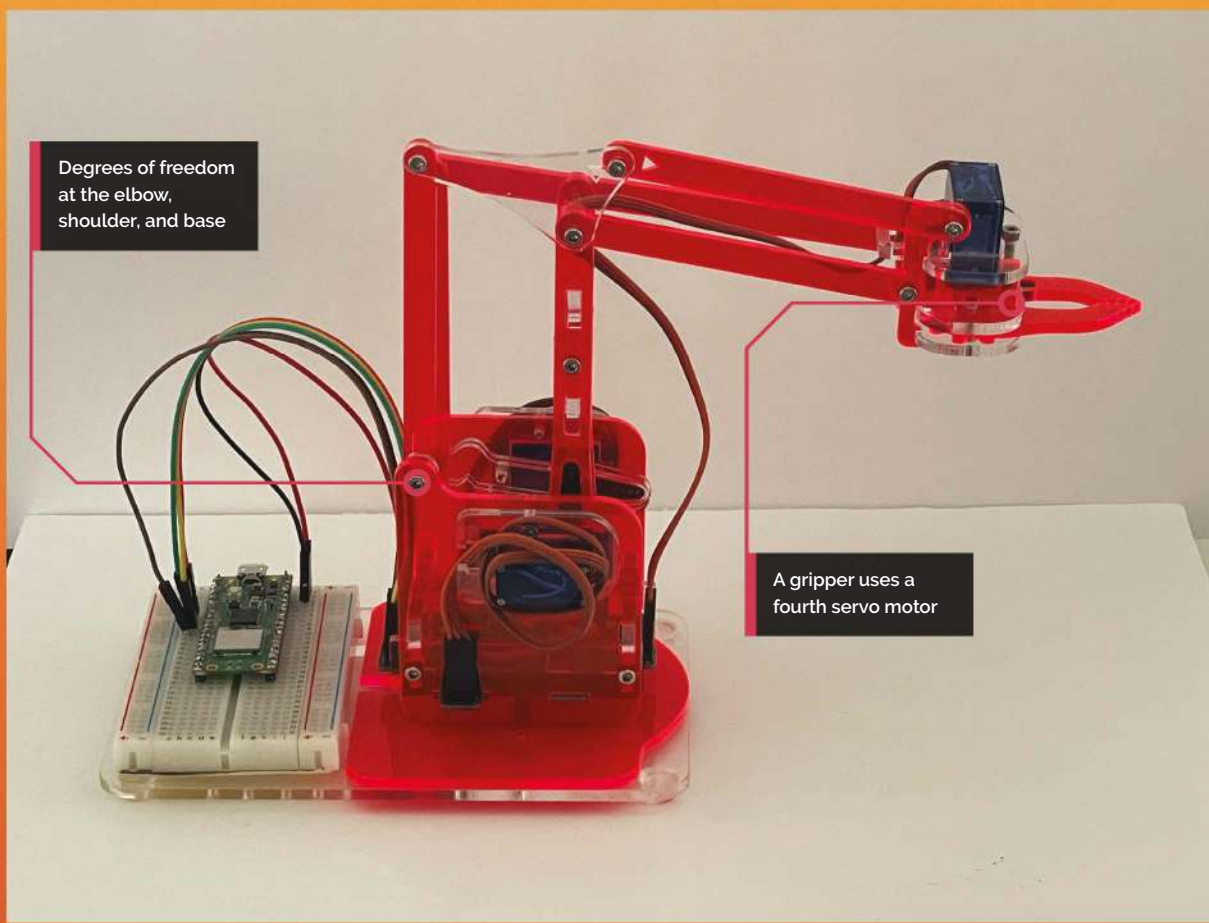
MAKER

Has a home robot lab, making robots with Raspberry Pi and Pico as Orionrobots. Author of *Robotics at Home with Raspberry Pi Pico*.

orionrobots.co.uk

YOU'LL NEED

- MeArm Maker **magpi.cc/mearmmaker**
- Breadboard **magpi.cc/breadboardhalf**
- Jumper wires **magpi.cc/jumperwires**
- Laptop, tablet, or smartphone with Wi-Fi
- A small cross-head screwdriver
- Thonny



Servo motors allow your Raspberry Pi Pico to control articulated robots, moving their joints to specific places.

In this tutorial, you will learn where servo motors can be used and how to connect a servo-driven robot arm to a Raspberry Pi Pico W.

You will learn how to use the asyncio library (**magpi.cc/asyncio**) to control servo motors together, making articulated robotics possible.

You will then control this robot arm from a web page or smartphone.

01 What are servo motors?

Articulated robots use servo motors to control their joints. Servo motors combine a motor, a gearbox, and control circuits. They are usually limited to 180 degrees of motion.

A servo motor uses closed circuit control and is controlled with pulse-width modulation or 'PWM', where the length of pulses carries position information.

A Raspberry Pi Pico can output these signals, with 65,536 possible values. Servo motors only respond to a range of 8000 values, with very short pulses 50 times per second. The servo finds the difference between its input and output position, then moves the motor to reduce the difference.

02 About the MeArm

The MeArm is an inexpensive, laser-cut robot arm kit from Mime Industries (**mime.co.uk**). MeArm is designed for hobbyists and learners to build and program. Mime Industries has MeArm models with electronics designed for Raspberry Pi, Arduino, and micro:bit. However, the MeArm Maker is a model which allows connection to any controller, and will work well with Raspberry Pi Pico.

Alternative robots

The techniques in this tutorial can be used for other articulated robots like Petoi Bittle X, the Hiwonder TonyPi, or for you to build a hexapod or robot tentacle!!

Degrees of freedom

When talking about segmented robots, like this arm, we use the term 'degrees of freedom', or DOF, to count the number of places or axes segments can move.

The MeArm uses four servo motors which allow it to move in three axes and pick things up. The three degrees of freedom are a rotating base, a shoulder joint, and an elbow joint. The axes are controlled by servo motors.

03 Preparing Pico W

Before we can interact with the arm, we need to prepare Raspberry Pi Pico W. First you will need to solder headers onto it, unless you bought a model with pin headers on board. See **magpi.cc/soldering** for help with soldering.

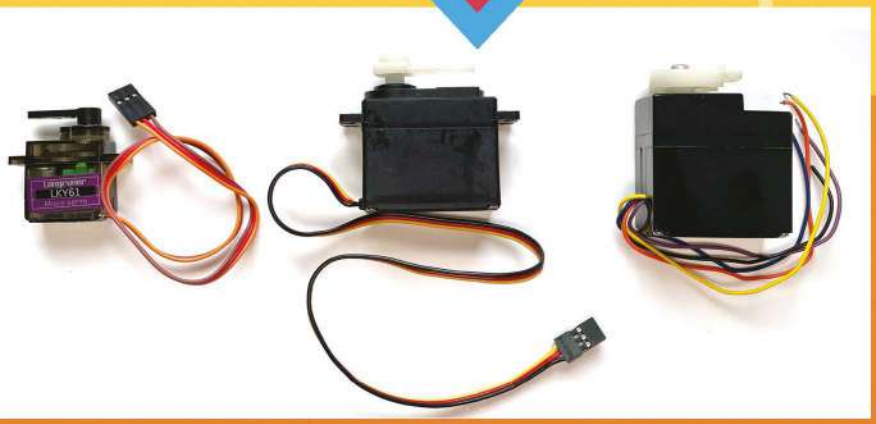
> "The MeArm is an inexpensive, laser-cut robot arm kit"

You will also need the current version of MicroPython for Raspberry Pi Pico W. Download the latest UF2 image from **magpi.cc/micropython**. Hold the BOOTSEL button while connecting Pico W to a computer via USB; you can then drag and drop the UF2 image onto Pico W on the desktop.

04 Wiring servos to Pico W

The servo motors will need to be configured before assembling the arm. Unplug Raspberry Pi Pico W from the USB port. Each servo motor connects to one of the 3-pin connectors on the MeArm board, with the orange or yellow wire going to the pin marked O or Y on the board.

The 6-pin connector is wired to the Raspberry Pi Pico W. Plug the Pico W into a breadboard, and then run a jumper from each labelled connection on the 6-pin connector to the corresponding breadboard row for the label on the Pico W.



A selection of servo motors. They come in different sizes to handle larger tasks. The MeArm robot uses tiny ones

05 Controlling servo motors with MicroPython

Plug in the Pico W, then open Thonny. Connect Thonny to the Pico W, and start the MicroPython Shell.

Type the following lines:

```
import machine
grip = machine.PWM(machine.Pin(4, machine.Pin.OUT))
grip.freq = 50
grip.duty_u16(5000)
```

5000 corresponds to the middle position for the motor. You can try other values and should see the motor spin. However, ensure it is at position 5000 before finishing this step.

Repeat this for the other three servo motors on pins 5, 6, and 7. Avoid turning servo motors by hand, as this is likely to break gears inside them.

06 Assembling the MeArm Maker

With the Pico still powered, follow the MeArm instructions linked below for attaching servo horns while the servo motors are in the middle position. You can play with servo motor positions here, but ensure you return them to position 5000 before assembling.

Next, unplug the USB connection to the Pico, then unplug the servo motors from the MeArm board. Leave the connections between the MeArm board and Pico W. Building the arm is best explained by the MeArm documentation, which you can find at **magpi.cc/buildingmearm**.

07 Controlling arm movements

The code to control PWM for servo motors is in a Servo class in **mearm.py**. It's helpful to see the servo motion as angles. We'll make the middle position 0, so the range is +/- 90.

The middle duty cycle value is 5000 for Pico MicroPython – we'll call this `PWM_MID`. If we take +/- 4000 as the range from this position, we can divide 4000 by 90 to get a `DEGREES_TO_PWM` scale factor.

We can then multiply an angle by this scale factor and add the result to the `PWM_MID` to get the duty cycle.

08 Making an arm from Servos

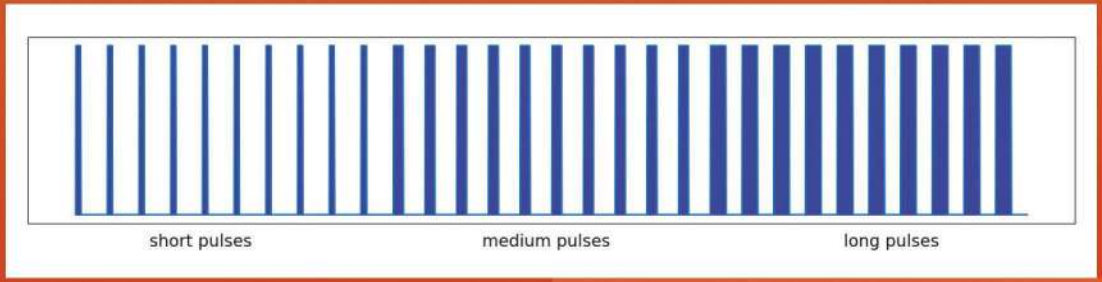
The Arm class uses the Servo class for each joint. The Arm class has default pin numbers corresponding to the wiring, but with different pin numbers, it allows two or more arms!

Load **mearm.py** in Thonny and hit the play button. You'll return to the MicroPython Shell.

Type `arm.base.set_angle(-30)`, to move the arm base to -30 degrees. You can try this with every joint. Find the limits by increasing numbers out from the 0 position in 5-degree steps until the joint doesn't move again, then go back a step. Don't leave a joint past these limits in order to avoid damage.

09 Smoother motion

The servos are jumping, but we can make them smooth. In `Servo.move`, we control speed



This chart displays PWM (pulse-width modulation) signals. Shaded parts show a high signal. Shorter pulses pull the servo anticlockwise, mid pulses go to the middle, and long pulses go clockwise

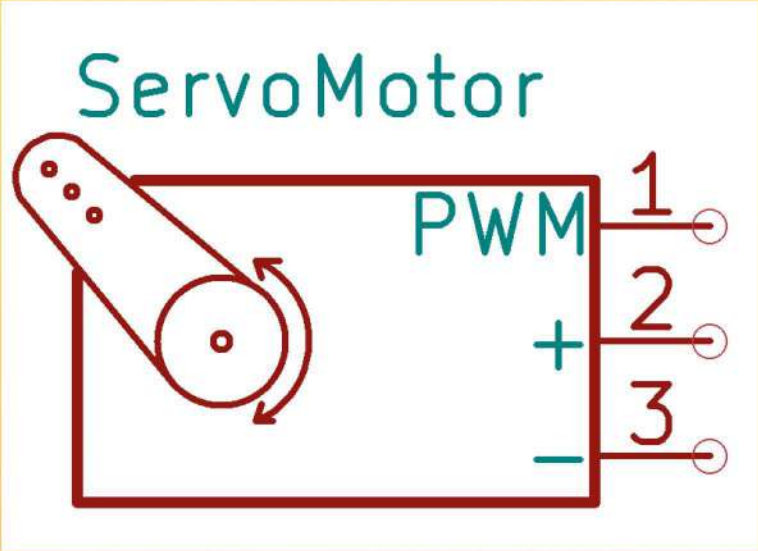
and smoothness with a time in seconds and with a step count.

The Servo class stores a `current` position; `Servo.move` sets a target and gets the difference between this and the `current` position.

`Servo.move` divides the time by steps for `step_time` and divides the movement difference by steps for `step_size`.

It loops over the steps, adding the `step_size` to the `current` position, moving the servo to this new position, and sleeping `step_time`.

In the Shell, type `do(arm.elbow.move(30))` to see the elbow move smoothly to 30 degrees.



A servo usually has three pins: '+' for voltage, '-' for ground, and a yellow or orange PWM input signal

10 Moving joints together

Moving joints together is more interesting. We used a `do` helper around `Servo.move` because `move` is asynchronous. When it encounters the `await` instruction, other code can run, enabling us to do multiple tasks at the same time.

The `move_together` method in `Arm` sets up a movement task for each specified joint position, and then uses `await uasyncio.sleep(seconds)` to wait for all the move tasks to complete. The tasks all get time to run, so things can move together.

> "It's helpful to see the servo motion as angles"

Try `do(arm.move_together(base=40, shoulder=20, grip=30, base=50))` to see everything move at the same time! Perhaps we could do something else?

11 Connecting to the network

We can connect our Pico W to a Wi-Fi network. Add a **secrets.py** file on the Pico with your SSID and Wi-Fi key.

```
SSID = "<network name>"
PSK = "<network password>"
```

The top of the file **web_arm.py** imports these secrets and uses them to connect to your Wi-Fi. We will also be using the microdot web server for MicroPython. You can install this from Thonny with Tools > Manage packages. Search for microdot, then click Install.

When you run **web_arm.py**, it will look for a connection. Once it has connected, it will print the network address of your Pico W to the console.

12 Building arm control endpoints

The microdot web server lets us build tiny endpoints to control our arm. There is a microdot app object, which lets us set a link between an incoming request path and a Python function, for example linking `/set_grip/<position>` to the function `set_grip`. The `<position>` part specifies a position parameter which the function can use and pass to the arm.

These functions are also in async, so the server can handle requests while the arm is still moving. Upload **secrets.py**, **mearm.py**, and **web_arm.py** to Pico W, then click the run button on **web_arm.py** to start the server.

13 Trying the endpoints

Point a web browser at the address you were shown in Step 11 plus a movement to make. For example: **http://192.168.0.136/set_elbow/30** to move the elbow to 30 degrees.

You do not have to wait for a previous movement to finish before sending more movement instructions to the robot. It will not behave correctly if you try to move the same motor twice before it's finished, though. This is OK, but it's not the most enjoyable way to interact with the robot.

The techniques in this article can be used to control the ten servo motors on the Petoi Bittle X robot dog

There's another endpoint in **web_arm.py**, named `/` and connected to an `index` function which serves a page to control the arm.

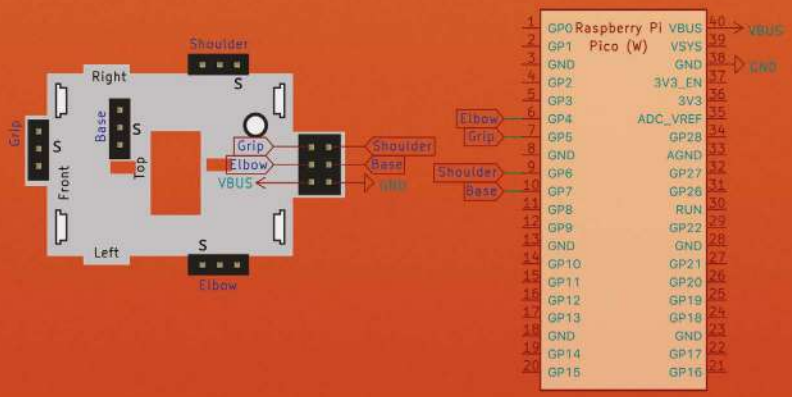
14 Putting sliders on a web page

The index function loads a **control.html** file and sends this back. The body of **control.html** has a title, and four `input type="range"` elements. Each range input lets you specify the minimum, maximum and starting values.

The datalist element `snap_to_zero` is used for the list property of the ranges to draw markers at 0.

The page has a style section, `overflow: hidden`, that stops the page being accidentally scrolled on a phone, The width and height settings make the sliders fit well on any screen, along with the viewport setting in the page head.

If you go to the address of your Pico W, you should see the four sliders.



Wiring the MeArm to Raspberry Pi Pico W. Wires go between labels with the same name

15 Connecting the sliders

In **control.html**, it loads `jquery`, a JavaScript library designed to help with handling page events and sending data from a page.

When the document is ready, our JavaScript searches for the input elements, and links on change handlers for each of them.

Focusing on the shoulder element, when this input is changed, the function will send a GET

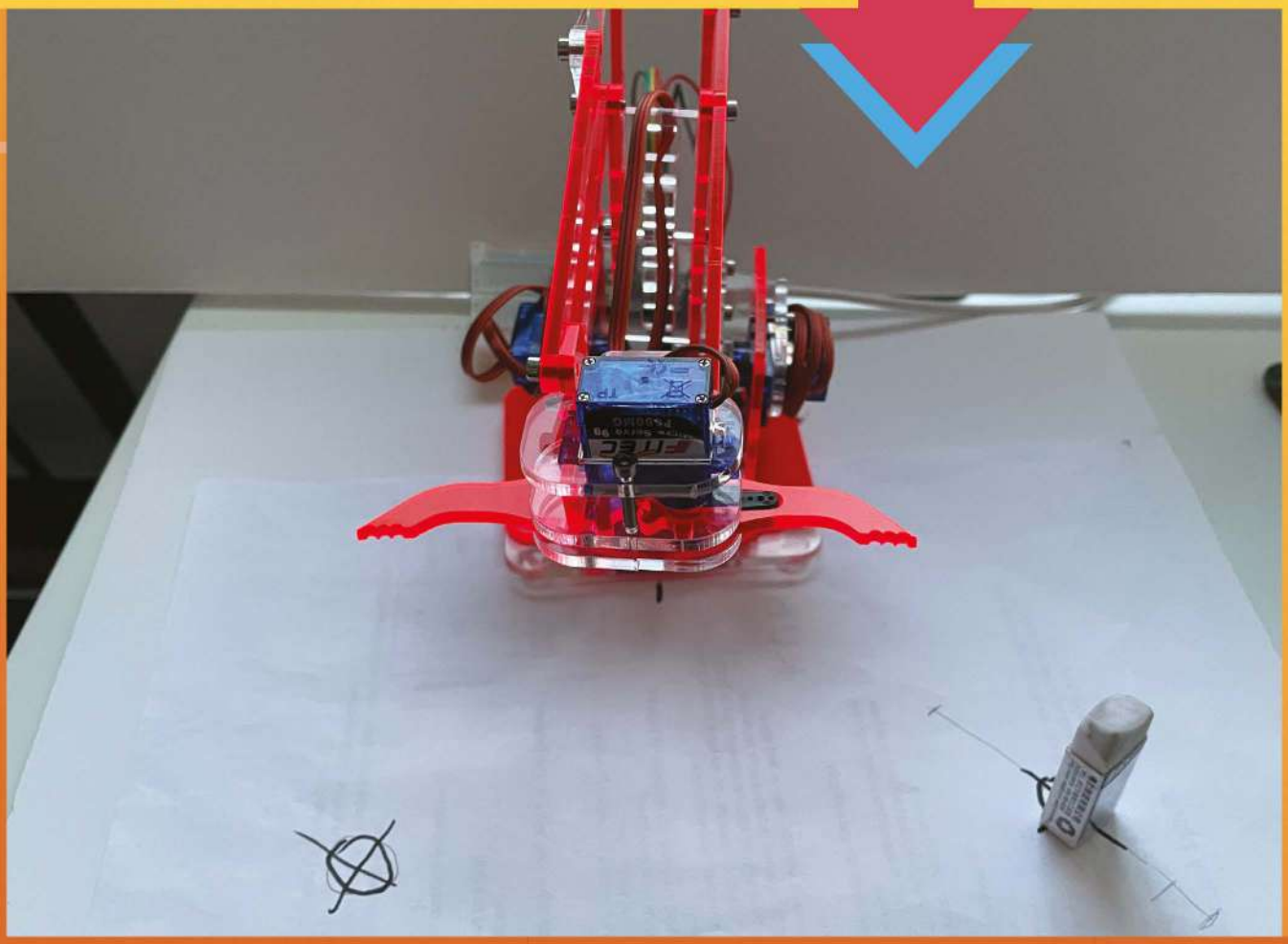
> You can drag sliders around to move the arm positions

request to the **web_arm.py** server. This goes to **/set_base/** and combines it with the value of the shoulder element. The change event happens when an input on the page is moved, and the finger or mouse button is lifted. This means you can drag sliders around to move the arm positions. You can connect with a smartphone to drive this too!

16 Recording steps

At the bottom of the page, there are buttons to add movement steps. In the HTML, they have onclick handlers to call into **web_arm.py**.

When the 'add step' button is clicked, **web_arm.py** collects the current position of each servo motor, and stores this set of positions in the variable `step_list`. Then, when 'run steps' is clicked, an async task calls `move_together` with each set of positions, playing the whole list back.



There are also save/load buttons. These will store the steps in a JSON file on Pico W, so you can load steps back later.

17 Fun with recorded steps

Put the robot on the long side of a sheet of paper. Draw a line in front of the base to align the arm's base. Draw a mark between 30 and 60 degrees to the left and 100 to 140mm from the base of the arm, and another mark to the right.

Place a light object, like a small eraser, on one mark. Using the web interface, open the grippers, rotate the arm to face the object, then move the arm so the object is between the open grippers. Press the 'add step' button. Close the grippers, and press 'add step'.

Now set the elbow, base, and shoulder joints to 0 and press 'add step'. Then move the base, elbow, and shoulders to put the object on the other mark and add a step. Open the grippers, then add a step. Now move the elbow and shoulders up, and add a step. Finally, set the elbow, base, and shoulder joints to 0 and press 'add step'.

If you put the object back on the first marker, and press the 'run steps' button, the robot will now automatically pick up the object and move it to the other marker. You can use the 'save steps' and 'load steps' buttons to save this for demonstrations!

▲ Setting out markers for the arm to pick up and move objects to

18 What to do next?

Connecting the gripper servo motor power through a current sensor, like an INA219, means you could detect when the grippers are holding something by closing them until current goes above a threshold.

Articulated robots use inverse kinematics to calculate servo angles based on a desired position. Adding this would make it easier to position the arm from the controls.

You could add a distance sensor to the front of the arm, move the base to scan for the object, and combine this with inverse kinematics to position the grips for an arbitrarily placed object.

mearm.py

DOWNLOAD THE FULL CODE:
magpi.cc/mearmpy

Language: **MicroPython**

```
"""
import sys
del sys.modules["mearm"]
from mearm import arm, do
# tests

do(arm.move_together(base=30, shoulder=40, elbow=30, grip=20))
do(arm.move_together(shoulder=0, elbow=-30))
"""
import machine
import uasyncio


PWM_MID = 5000
PWM_FREQ = 50
DEGREES_TO_PWM = 4000 / 90


class Servo:
    def __init__(self, pin):
        self.pwm = machine.PWM(machine.Pin(pin, machine.Pin.OUT))
        self.pwm.freq(PWM_FREQ)
        self.current = 0

    def set_angle(self, angle):
        self.pwm.duty_u16(int(PWM_MID + ( angle * DEGREES_TO_PWM)))

    async def move(self, position, seconds=1, steps=100):
        step_time = seconds/steps
        step_size = ( position - self.current) / steps
        for n in range(steps):
            await uasyncio.sleep(step_time)
            self.set_angle(self.current)
            self.current += step_size


class Arm:
    def __init__(self, elbow_pin=4, grip_pin=5, shoulder_pin=6, base_pin=7):
        self.grip = Servo(grip_pin)
        self.elbow = Servo(elbow_pin)
        self.shoulder = Servo(shoulder_pin)
        self.base = Servo(base_pin)

    async def move_together(self, base=None, shoulder=None, elbow=None, grip=None, seconds=1, steps=100):
        if base is not None:
            uasyncio.create_task(self.base.move( base, seconds=seconds, steps=steps))
        if shoulder is not None:
            uasyncio.create_task(self.shoulder. move(shoulder, seconds=seconds, steps=steps))
        if elbow is not None:
            uasyncio.create_task(self.elbow. move(elbow, seconds=seconds, steps=steps))
        if grip is not None:
            uasyncio.create_task(self.grip. move(grip, seconds=seconds, steps=steps))
        await uasyncio.sleep(seconds)


arm = Arm()
do = uasyncio.run
```

web_arm.py

Language: **MicroPython**

```
import machine
import uasyncio
import json

from microdot_asyncio import Microdot
from mearm import arm

import network
from utime import sleep

from secrets import *

def wifi_connect():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(SSID, PSK)
    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)
    print(wlan.ifconfig())

app = Microdot()
step_list = []

@app.route('/')
async def index(request):
    with open("control.html") as content:
        return content.read(), 200, {'Content-Type': 'text/html'}

@app.route('/set_grip/<position>')
async def set_grip(request, position):
    uasyncio.create_task(arm.grip.move(int(position)))
    return 'Moving'

@app.route('/set_base/<position>')
async def set_base(request, position):
    uasyncio.create_task(arm.base.move(int(position)))
    return 'Moving'

@app.route('/set_shoulder/<position>')
async def set_shoulder(request, position):
    uasyncio.create_task(arm.shoulder.move(int(position)))
    return 'Moving'

@app.route('/set_elbow/<position>')
async def set_elbow(request, position):
    uasyncio.create_task(arm.elbow.move(int(position)))
    return 'Moving'

@app.route('/add_step')
async def add_step(request):
    state = (arm.base.current, arm.shoulder.current, arm.elbow.current, arm.grip.current)
    step_list.append(state)
    return 'Added step'

async def run_steps():
    try:
        for step in step_list:
            await arm.move_together(*step, seconds=1)
    except:
        print("Steps were:", step_list)
        raise

@app.route('/run_steps')
async def handle_run_steps(request):
    uasyncio.create_task(run_steps())
    return 'Running steps'

@app.route('/save_steps')
async def handle_save_steps(request):
    with open("steps.json", "w") as f:
        json.dump(step_list, f)
    return 'Saved steps'

@app.route('/clear_steps')
async def handle_clear_steps(request):
    step_list.clear()
    return 'Cleared steps'

@app.route('/load_steps')
async def handle_load_steps(request):
    global step_list
    with open("steps.json") as f:
        step_list = json.load(f)
    return 'Loaded steps'

try:
    wifi_connect()
    print("Starting app")
    app.run(port=80)
except KeyboardInterrupt:
    machine.reset()
```

State of Play: Game Engines for Raspberry Pi

Develop anything from text adventures to 3D worlds with four sophisticated game engines

MAKER

K.G. Orphanides

K.G. is a software preservationist and developer who's written middling text adventures for fun and done despicable things to 3D engines for profit.

magpi.cc/owlbear

You don't need a fancy integrated development environment (IDE) or graphical engine to create fantastic computer games, but they really can help. With mature choices of 64-bit operating system for Raspberry Pi and enthusiasm among developers for releasing software to support Linux on aarch64 hardware like Raspberry Pi 4's, game engine support for the platform has expanded significantly since we last explored our options for game development. We're going to highlight four of the most exciting game engines you can use to make your own games on Raspberry Pi today.

Godot 3.5

magpi.cc/godot35

64-bit? Yes
32-bit? Yes

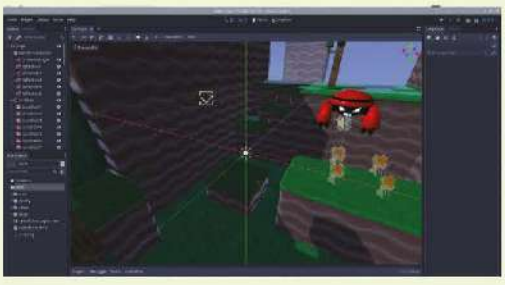
Godot is a free, open-source game engine for 2D and 3D games. In fact, it's one of your few fully functional choices for 3D games development on Raspberry Pi, and includes a full IDE in the vein of Unity and Unreal Engine. That said, the editor does chug a bit with a reasonably busy 3D scene loaded in, so you might want to stick to 2D, or only the simplest 3D scenes, if Raspberry Pi is your primary development system.

We recommend the Unofficial Godot for Raspberry Pi port. On the downside, this means you're limited to Godot 3, rather than the slightly snazzier Godot 4. The latter can also be built for Raspberry Pi, but there isn't currently a consistent deployment path for this, so it's best to stick with the Unofficial Godot port of 3.5 for stability. Both Godot 4 and the Raspberry Pi port of Godot 3.5 allow you to target your game builds at Raspberry Pi.

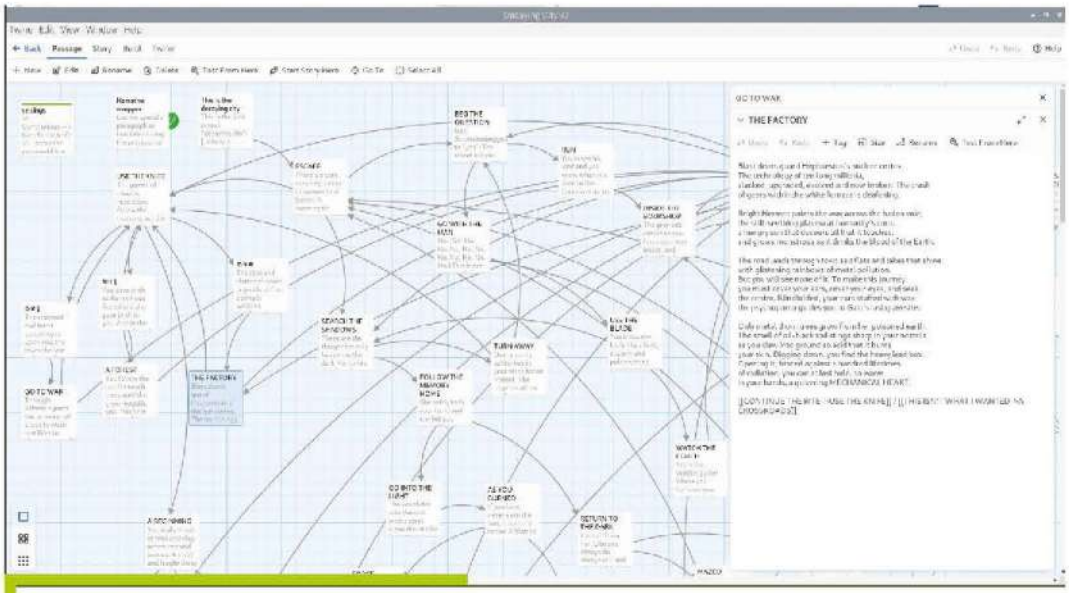
By default, Godot uses its own GDScript language, a high-level, object-oriented language with Python-like syntax. It's very quick to learn and is, in fact, your only option in this Raspberry Pi port, although other versions of Godot offer optional support for C#, C, and C++. Godot has comprehensive documentation with some fantastic tutorials to get you started – just make sure you're following the docs (**magpi.cc/godotdocs**) for version 3.5.

Install it:
Download the latest release for your Raspberry Pi from **magpi.cc/godotrpi**.
Unzip it and run the relevant version.

Notable works:
Brotato, Cassette Beasts, Cruelty Squad, RPG in a Box.



Godot is a full 2D/3D game engine and integrated development environment. Everything works, though 3D scenes can run slow when you play them



Twine allows you to create complex choice-based games. Minimal programming is required, but you can create surprisingly complex routines using story formats like Harlowe

Twine

twinery.org

64-bit? Yes
32-bit? No

Twine is one of the most approachable ways of making a text game, allowing you to create choice-based interactive fiction and choose-your-own-adventure narratives. It's based on HTML and JavaScript, making it easy to publish your games online and even extend the features of the game engine, which by default supports sound, images and video, just as HTML5 does.

Twine has made its mark in the world of interactive fiction over the years, has become one of the most popular engines in the IFComp contest, been used by narrative designers and game writers as a prototyping and plotting tool in larger projects, and had a significant presence in the LGBTQ+ games movement of the 2010s. We've been running Twine on Raspberry Pi for years, starting with Raspberry Pi 3 running a 32-bit operating system. Things have moved on since our 2016 tutorial, way back in issue 64. Although our guide to developing for Twine's Harlowe language is still relevant to modern versions, Twine's IDE has seen some welcome polish in recent years.

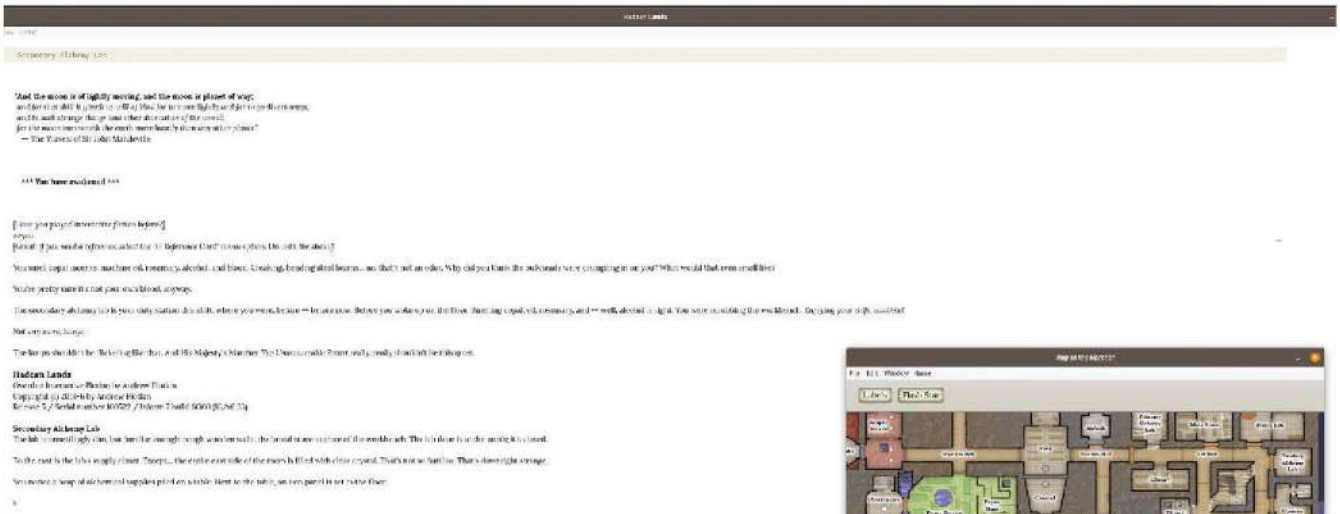
There are four built-in story formats: Chapbook, Harlowe, Snowman, and SugarCube. Your choice of story format determines the appearance, behaviour, and syntax of your Twine creations. Harlowe is the beginner-friendly default format and our recommendation for anyone starting with Twine, as it's both easy to use and flexible enough to allow you to easily bolt on additional functionality.

SugarCube is the continuation of Twine's first story format, and benefits from a mature macro language; Chapbook is a more 'literary' and generally book-like story format, great for making traditional choice-based books, and good for authors without any programming knowledge. Snowman is a bare-bones but highly customisable story format designed for developers who are already comfortable with JavaScript and CSS.

Whichever story format you choose, Twine's development interface presents you with a series of interlinked boxes, each containing a passage of text that you can think of as a paragraph, a page, or a location in your adventure game, Your finished game will be spat out as an HTML5 plus JavaScript experience that can be played via a web browser – just upload it to your website and share the link. A number of tools and wrapper techniques exist to turn your web-based game into a stand-alone offline experience, although that's not the primary focus of the engine. Indie game distribution platform itch.io directly supports Twine games.

Install it:
Raspberry Pi users running a 64-bit operating system should go to **twinery.org**, click 'Download desktop app', select the Linux-arm64 version, unzip it, and run the file named **twine** from a terminal with `./twine`.

Notable works:
Depression Quest, This Book is a Dungeon, Horse Master: The Game of Horse Mastery, Black Mirror: Bandersnatch (development).



▲ Inform 7 is a mainstay of the creative interactive fiction community and has also been used for commercial titles such as Andrew Plotkin's Hadean Lands

Inform 7

magpi.cc/inform

64-bit? Yes
32-bit? No

Inform is one of the best-known interactive fiction languages, and version 7 is the pinnacle of its development: a natural-language text adventure authoring system in English; to create a space for players to interact with, you type something like 'The Shed is a room,' followed by its description. Interactive objects and characters are defined in a similar way. Open-sourced in 2022, Inform 7 allows writers to use the familiar milieu of words to develop games with complex functions and rich interactivity.

The Inform 7 IDE is an immensely useful tool. It's set up with two pages, containing your game's script on the left..

Inform 7 can output Z-Machine story files, the de facto interactive fiction community standard, compatible with the virtual machine used by legendary text adventure developer Infocom in games such as Zork; or Glulx files, which are based on Z-Code but without the older standard's 64kB memory limitation. Unless you're particularly committed to 40 years' worth of backwards compatibility, we recommend opting for Glulx. Glulx interpreters for Raspberry Pi include Gargoyle, ScummVM, and the browser-based Parchment interpreter. To get started playing Glulx games, open a terminal and type `sudo apt install gargoyle-free`.

If you fancy something a bit more traditional, or if you're an interactive fiction developer working in a language other than English, Inform 6 (**github.com/DavidKinder/Inform6**) is a viable alternative, with a compiler that runs on Raspberry Pi. However, Inform 6 lacks the integrated development environment of its latest successor. The IDE, with its area maps and comprehensive integrated authoring guide, helps to set Inform 7 apart from both its predecessors and many of its rivals.

Install it:

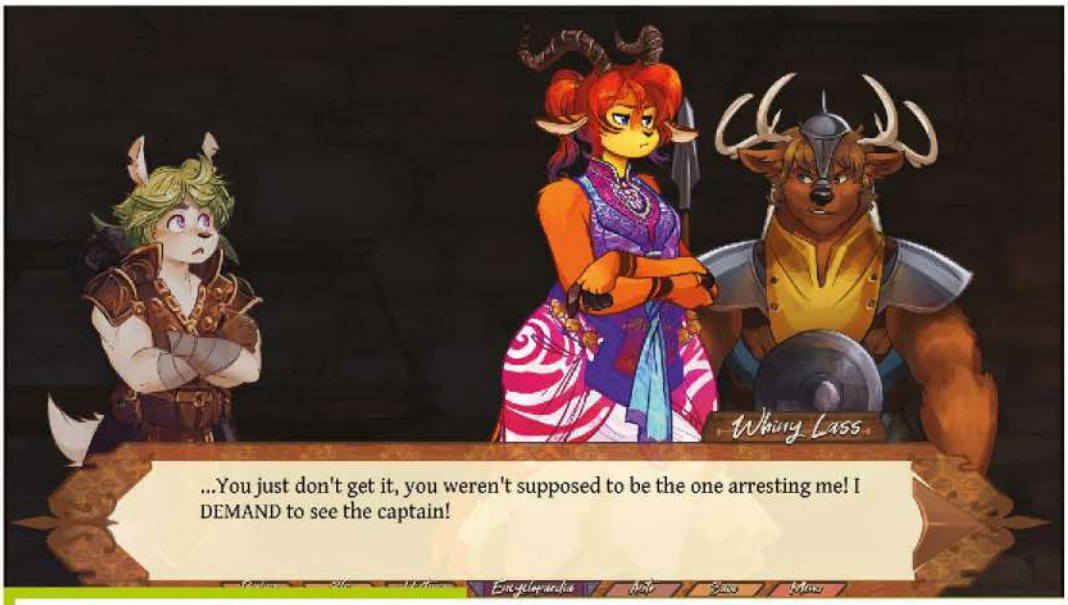
```
sudo apt install flatpak
flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
flatpak install inform
```

Select **app/com.inform7.IDE/aarch64/stable** (option 2 at the time of writing) and type **Y** to install it.

Approve any other required changes to install dependencies by typing **Y**.

Inform 7 will be added to the main menu's Programming section.

Notable works:
Anchorhead, Counterfeit Monkey, Hadean Lands.



Ren'py is the engine underlying numerous very polished commercial visual novels, such as Maxi Molina's The Hayseed Knight

Ren'py

renpy.org

64-bit? Yes (run 32-bit version)
32-bit? Yes

If you've played a modern indie visual novel (VN), it was probably made using Ren'py, a specialist game engine built on top of Python, with a range of almost-natural-language functions designed to do the kind of things you'd expect in a visual novel: load a background scene, show a character, place some building, and display narrative text or dialogue. It's elegant, and the core functions have been kept simple, aided by a superb quickstart guide. This makes it a particularly good choice for artists who wish to start developing their own games.

But you can extend its normal capabilities and features with your own Python scripts, invoked directly from within Ren'py. For example, if you want to add health and combat mechanics to your VN, you'll want to turn to Python. In fact, Python makes Ren'py so flexible that developers are sharing code for minigames ranging from go and solitaire to fishing and lockpicking.

With Ren'Py 8, the entire engine has moved to Python 3, fixing a number of longstanding issues, such as poor handling of accenting and non-Latin characters. Even though Ren'py's official documentation is written to manage your expectations of the engine's ability to run on Raspberry Pi, we found that Ren'py runs in full, with no issues, on Raspberry Pi 4 computers running either 32-bit or 64-bit Linux operating systems, thanks to a dedicated ARM Linux build.

Raspberry Pi 3 can run most Ren'Py games, allowing you to develop for the platform, but may struggle with Ren'py's graphical launcher, depending on your OpenGL settings. This isn't required to develop your games, however, as you'll be spending most of your time using your preferred editor. We use VS Code with the Ren'Py Language extension installed. You can use Ren'Py to develop kinetic visual novels – those that involve no interaction from the reader beyond moving to the next page – as well as VNs with choice-based gameplay and branching dialogues, and even those that incorporate elements of other genres, such as RPG-style combat or the minigames we mentioned above.

One last thing to bear in mind when looking for Ren'py games and sample code is that it's hugely popular for creating adults-only erotic visual novels, and even works that don't feature any sexual content often touch on mature themes. Authors are generally great about providing appropriate warnings about sexual and mature content, but it's worth bearing in mind if you're looking for examples to share with younger developers.

Install it:
Go to **renpy.org** and click on the button to download the version you want – at the time of writing, the latest version is Ren'Py 8.1.0.

Select the 'Download ARM Linux (Raspberry Pi and Chromebook) SDK' link, unzip it, and run `renpy.sh`.

Notable works:
The Hayseed Knight, Loren The Amazon Princess, Long Live The Queen, BBQ DAD.

Build a classic text adventure in Inform

Get started in interactive fiction development

Top Tip

Sentence structure

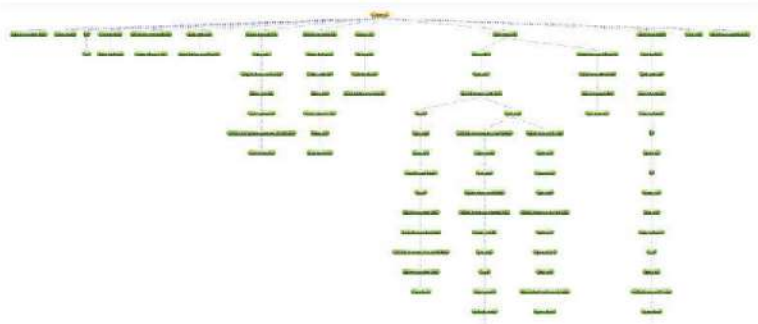
Although we've organised our code in successive lines for ease of reading, Inform will cheerfully accept paragraphs.

Inform can be used to create very traditional quest-based text adventures, sprawling and intricate works of genre fiction, interactive theatre, and experimental art games. You can add graphics and sound, and a range of extensions (**magpi.cc/intfic**) have been released by the community to add features and functionality. As an introduction to Inform 7's language and capabilities, we're going to create a simple escape room puzzle, with a couple of locations and a handful of objects. In The Shed of Mystery, the player's going to have to work out how to escape a garden shed. You will need a 64-bit installation of Raspberry Pi OS to run the Inform 7 flatpak.

01 Your new adventure

Open Inform 7 and click 'Start a new project'. Click Next to create a new story. Select or create a directory for your project to live in. Enter your game's name and your own, then click Next. Check your project settings, and click Apply.

Inform's main interface opens with an editor on the left and a documentation pane on the right. Tabs on both let you switch between different content, including extensions, settings, and a map of your previous play-throughs. A play button just above the right-hand pane allows you to play your game. Note the circle arrow next to it: this will replay every command you typed in your previous play-through, allowing you to immediately see the impact of changes to your code.



Inform's 'skein' records every sequence of actions you've played through, illustrating paths through your game

02 Create a room

We'll start by creating and describing a room. Type the following:

The Shed is a room. 'A cluttered wooden shed, lined with shelves and smelling faintly of creosote.'

The first room that appears in your project is where your player will start the game by default, but you can always override this with a line reading 'The player is in [the room you wish them to start it]' at the beginning.

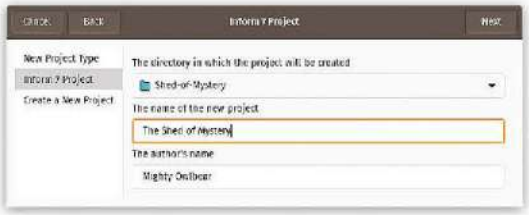
To help with larger games, you can also group Rooms into Regions, and create Scenes, which can include scripted actions or specific in-game time periods.

03 Your first objects: Supporters

To create an object in Inform 7, you start by naming it. Type:

A workbench is here. The workbench is a supporter. Instead of examining the workbench, say 'The old workbench is almost, but not entirely, free of mess.'

That's created an object called workbench and defined it as a 'supporter', an object which can



It's a good idea to give all your Inform projects their own directories, just to keep everything tidy

have other objects on top of it. Pay attention to that 'instead of examining' phrase. If we'd just given the workbench a description, as we did the shed, the workbench would be described in full as part of the description of any room that it's in. We only want to show a description when a player LOOKs at the workbench, so our 'instead of' construction provides a specific phrase for that.

04 Inside containers

'Containers' are objects that can contain other objects. Type:

A biscuit tin is a closed container on the workbench. The biscuit tin is openable. The biscuit tin contains a pin.

When you test that code by running the game, you'll be able to OPEN the biscuit tin and TAKE the pin. Now, on a separate line in the editor, type:

A red balloon is here. 'A red balloon is floating around.'

We've given the balloon a description, which, unlike the workbench, will be displayed as part of the description of any room that it's in.

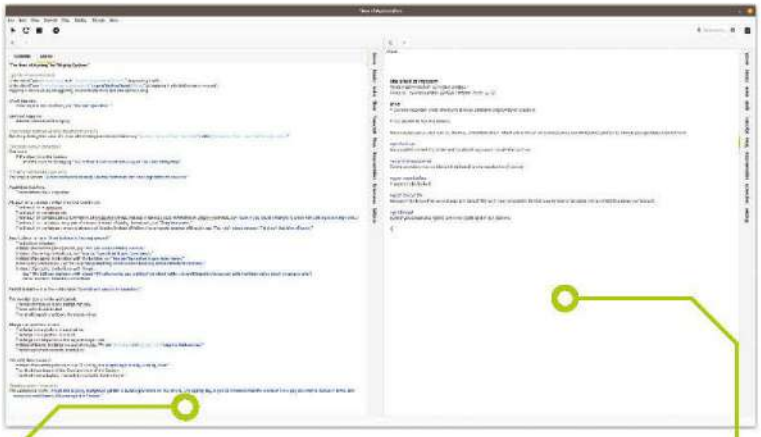
05 Extending Inform's verbs

We've got a pin and a balloon, but Inform doesn't know about popping. Let's change that. Towards the top of our code, we've created a comment – text between square brackets – that reads **[specify new command: pop]**. Below that, we've used an 'understand' statement to describe how our new verb works, and called the 'check' and 'carry out' rulebooks (**magpi.cc/informverb**) to make our 'pop' verb interact correctly with poppable items.

Returning to our red balloon's code, we can now make it poppable and put something inside it. Type:

A small brass key is in the red balloon. 'A small and very shiny brass key.'

Instead of popping the balloon with the pin: say 'The balloon explodes with a loud POP, showering you in bits of shredded rubber. A small brass key drops out onto the floor with a small pinging sound.'; move the small brass key to the Shed.



Perhaps most helpfully, you can keep your code in view while playing through your game. There's also a toolbar button to automatically replay your last play-through, allowing you to change your code on the fly and see how it affects things, among other automated testing features

The Inform IDE's two panes allow you to have your code always visible while you use the second pane to read the extensive built-in documentation and code samples, display compilation errors, and manage settings and extensions

06 Create an exit

The player needs somewhere to escape to and a locked door to prevent them leaving. Type:

The Garden is a room.

It's best to organise rooms and their objects together, so we've put the Garden below everything to do with the Shed. Above it, type:

The shed door is a door. Instead of examining the door, say 'A scruffy door.' The shed door is east of the Shed and west of the Garden. The shed door is locked. The small brass key unlocks the shed door.

07 Winning the game

The player wins when they escape into the garden. At the very top of our code, we've specified a rule that runs every turn to see if they're there:

Every turn:
If the player is in the Garden:
end the story finally saying 'You're free! You have at last escaped The Shed of Mystery!'

Download our extended Shed of Mystery code (**magpi.cc/shedofmystery**) for two more keys to find, extract and use, as well as more items and descriptions. Play with your own shed and see what extra escape room challenges you'd like to add.

Top Tip

Rule of law

Inform 7's syntax is defined by 'rulebooks' such as 'check' and 'carry out'. See **magpi.cc/informverb2** for details.

Top Tip

[magic brackets]

Square brackets are variously used in Inform 7 to define a comment, style text, or as text replacements in text strings or rules.

shed-of-mystery-mini.txt

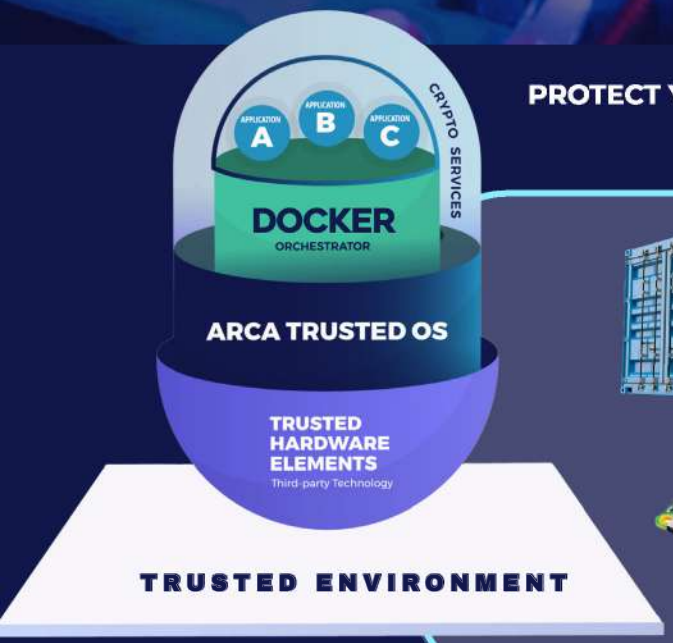
DOWNLOAD THE FULL CODE:
magpi.cc/shedofmystery

Language: **Inform 7**

```
001.  "Shed of Mystery Mini" by "Mighty Owlbear"
002.
003.  [check for our win condition]
004.  Every turn:
005.          If the player is in the Garden:
006.                  end the story finally saying "You're free! You have at last escaped
      The Shed of Mystery!"
007.
008.  [specify a new command]
009.  Understand "pop [something]  with [something preferably held]" as popping it with.
010.  Understand "use [something preferably held] to pop/destroy/burst [thing]" as popping it
      with (with nouns reversed).
011.  Popping it with is an action applying to one visible thing and one carried thing.
012.
013.  Check popping:
014.          if the noun is not a balloon, say "You can't pop that."
015.
016.  Carry out popping:
017.          remove the balloon from play
018.
019.  [Create a room and its contents]
020.  The Shed is a room. "A cluttered wooden shed, lined with shelves and smelling faintly
      of creosote."
021.
022.  A workbench is here. The workbench is a supporter. Instead of examining the workbench,
      say "The old workbench is almost, but not entirely, free of mess."
023.
024.  A biscuit tin is a closed container on the workbench. The biscuit tin is openable. The
      biscuit tin contains a pin.
025.
026.  A red balloon is here. "A red balloon is floating around."
027.  A small brass key is in the red balloon. "A small and very shiny brass key."
028.  Instead of popping the balloon with the pin: say " The balloon explodes with a loud
      POP, showering you in bits of shredded rubber. A small brass key drops out onto the
      floor with a small pinging sound."; move the small brass key to the Shed.
029.
030.  The shed door is a door. Instead of examining the door, say "A scruffy door." The shed
      door is east of the Shed and west of the Garden. The shed door is locked. The small
      brass key unlocks the shed door.
031.
032.  [Another room. Freedom! ]
033.  The Garden is a room. "A lush and slightly overgrown garden is bursting with life on
      this bright, late Spring day. A gentle breeze rustles the leaves of trees populated by
      a chorus of birds, and many-coloured insects flit amongst the flowers."
```

ARCA TRUSTED OS

A trust computing base to protect your containerized AI/ML and your data at the edge



PROTECT YOUR CONTAINERS AGAINST ATTACK PROPAGATION AND DATA COMPROMISSION

Docker Container

Rapberry Pi 4B

ARCA Trusted OS : A secure minimalist Operating System to host containers for your Raspberry Pi 4B

- Hardware Root of Trust with OTP or TPM 2.0
- Extended Secure Boot to verify the trustworthiness of the container OS
- Immutable file system to prevent unauthorized modifications
- Full Disk Encryption by default with key rotations
- Simple and central management of a fleet of edge devices
- OTA updates to maintain the OS with regular patches provided by CYSEC

TEST IT FOR FREE
OR VISIT
WWW.CYSEC.COM

CYSEC

CYSEC is an European data security company focused on data protection in hybrid and distributed architectures

All rights reserved - 2023 - CYSEC SA

Part 05

Build a Pico Keypad Mole Bop game

Create a game of Mole Bop using a 16-button, light-up keypad connected to a Raspberry Pi Pico using a port expander

MAKER

Stewart Watkiss

Also known as Penguin Tutor. Maker and YouTuber that loves all things Raspberry Pi and Pico. Author of *Learn Electronics with Raspberry Pi*.

penguintutor.com

@stewartwatkiss

Understand a way to expand your electronic circuits using I2C. Learn about how to interface with a port expander and see how this can be used for a large number of switches using a switch-array. The first circuit is based around an integrated circuit that you can build on a breadboard. Then, see how a similar technique is used for a 16-button, light-up keypad. Use the keypad to create a fun game of Mole Bop.

You'll Need

- 2 × button switches **magpi.cc/sanwa**
- MCP23008 port expander **magpi.cc/MCP23008**
- Pico RGB Keypad Base **magpi.cc/picokeypad**

01 Using a port expanders

Raspberry Pi Pico has 26 general-purpose input/output pins. For many projects that is more than enough, but sometimes we need more. For a 16-way keypad using a port for each switch and one for each LED, we would need 32 ports.

There are different ways to reduce the number of ports needed. To reduce the number of pins for a keypad, you could use a switch matrix which uses one port for each row and column and scan across them. An example matrix keypad is shown in **Figure 3**. A switch matrix can have a problem with ghosting when multiple buttons are pressed. Sometimes it's easier to use a port expander which increases the number of GPIO ports.

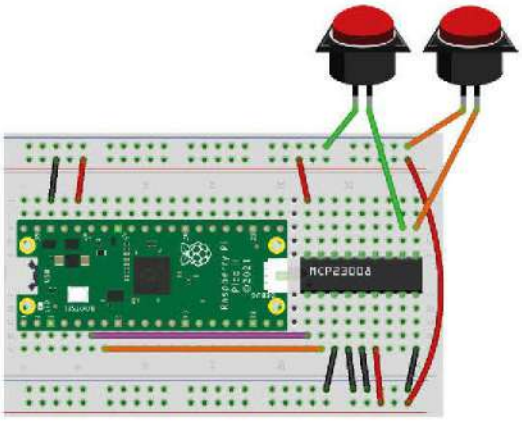


Figure 1 The MPC20008 is a port expander which can be used on a breadboard as a demonstration of how a port expander can be used

02 Using I2C

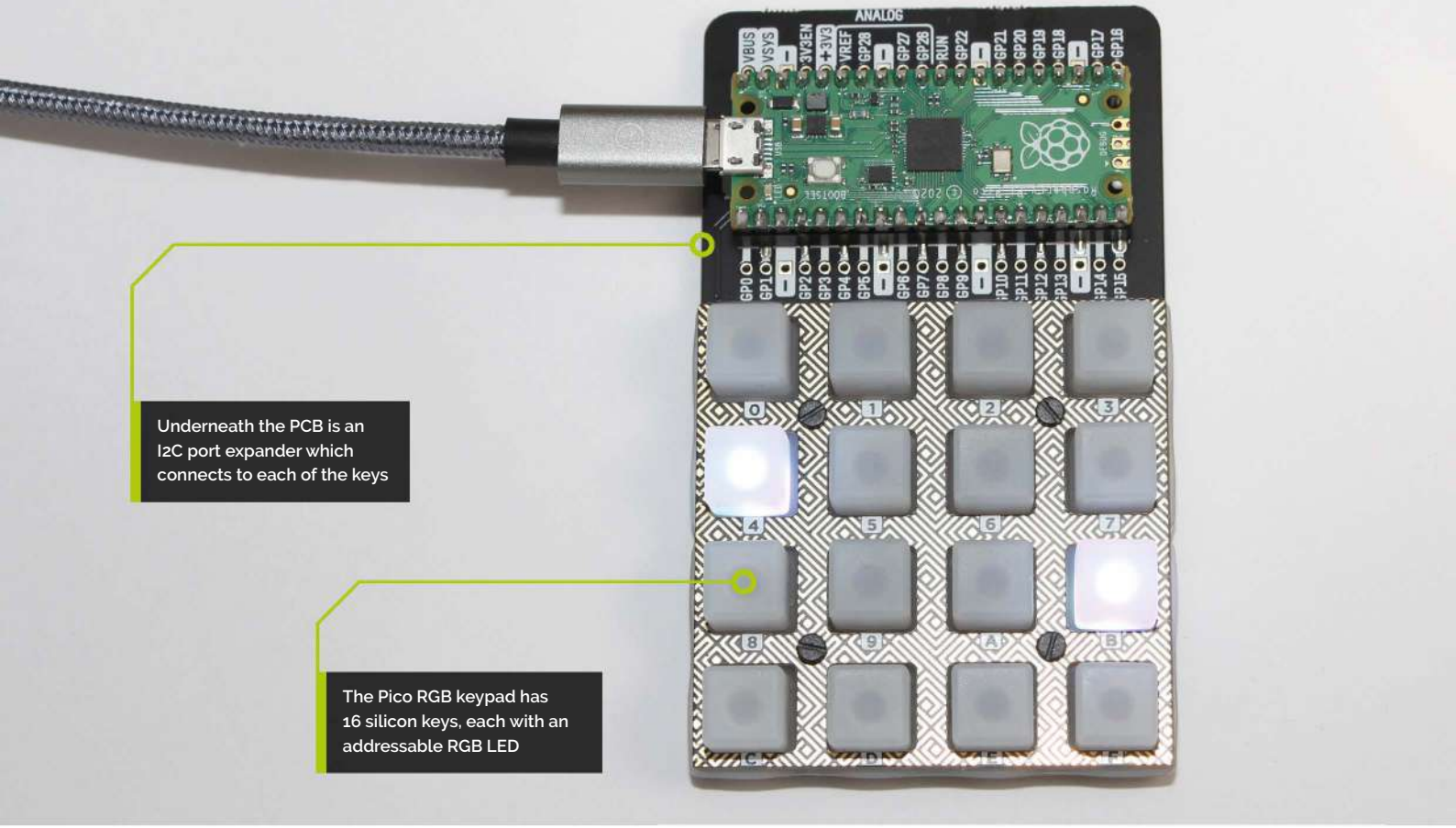
I2C is an abbreviation for Inter-Integrated Circuit, which is a protocol for communicating between microcontrollers and integrated circuits. It is a bus technology which allows multiple devices to be connected using only two wires; one for the data and the other for a clock to synchronise the devices. Each device has an address which is used to determine which device to talk to. An example is shown in **Figure 2**.

When using I2C, there needs to be pull-up resistors on the data and clock wires. These can use the internal pull-up resistors available within your Pico; if using devices over a greater distance, external pull-up resistors may be required.

03 Using a MCP23008 port expander

The MCP23008 is an 8-bit port expander which uses I2C to communicate with a Pico or other microcontroller. There are similar devices, such as the MCP23017, which have more ports, but using the smaller MCP23008 means that you can create a circuit on a single breadboard. The example circuit has just two buttons used as a demonstration. This is shown in **Figure 1**.

The address is configurable using pins 3, 4, and 5 on the IC. Connecting all those pins to ground gives an address of 0x20 (hexadecimal value). The


Underneath the PCB is an I2C port expander which connects to each of the keys
The Pico RGB keypad has 16 silicon keys, each with an addressable RGB LED

two wires between the MCP23008 and a Pico are the I2C clock and data signals.

04 Communicating with the port expander

You can use MicroPython to access all the extra ports, almost as easily as if they are local pins. To communicate between a Pico and the port expander, you first need to download the library from **magpi.cc/crankshawnzmp**.

After uploading the library to your Pico, you need to create an instance of I2C and use that to create an instance of the MCP23008 library. The following uses pins 4 and 5 on your Pico and address 0x20 on the port expander:

```
from machine import Pin, I2C
import mcp23008
i2c = I2C(0, scl=Pin(5), sda=Pin(4))
mcp = mcp23008.MCP23008(i2c, 0x20)
```

05 Reading the inputs

The pins can be configured as inputs using the `setPinDir` command. This takes the port number as the first argument and the value 1 for the second argument to indicate that it is used as an input (0 would be used for output).

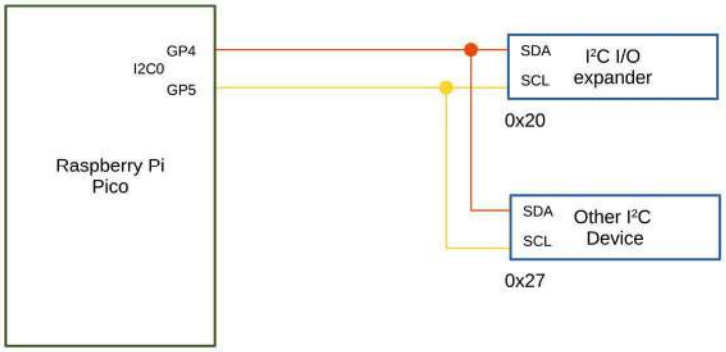


Figure 2 I2C is a bus protocol allowing multiple devices to be connected using two GPIO ports on Raspberry Pi Pico

```
mcp.setPinDir(0,1)
```

You can also enable pull-ups for the buttons using `setPullupOn`.

```
mcp.setPullupOn(0)
```

Once it is configured, you can query the status using `readPin`.

```
mcp.readPin(0)
```

This gives a 1 when the button is pressed, otherwise a 0.

Top Tip

Other I2C devices

There are many other devices which use I2C, including various sensors and output devices such as LCD screens.

06 Pico RGB keypad base

Using a port expander, you could connect that to buttons and LEDs to create a keypad. Alternatively, you could buy a Pico RGB keypad base which has already wired a keypad on a custom PCB with a 4×4 silicone keypad. The keypad uses a TCA9555 port expander on the PCB (**Figure 4**). This is like the MCP23008, but has 16 input/output ports and is a surface-mount package (better for PCBs, but not so good for breadboards). The chip even uses the same I2C address of 0x20. The PCB also includes an APA102 addressable LED for each key. These are like NeoPixels, but use a protocol based on SPI instead of the single data connection that NeoPixels use.

07 Preparing the keypad base

The base needs a little assembly for the buttons, then you plug a Pico using header pins. Instead of installing the drivers separately, Pimoroni provides a version of MicroPython with all the drivers pre-installed. Download the Pico file from **magpi.cc/pimoronipicogit**. Transfer the UF2 file to your Pico after pressing the BOOTSEL button during startup. The keypad and LEDs can be accessed by importing picokeypad and creating an instance of that class.

```
import picokeypad
keypad = picokeypad.PicoKeypad()
```

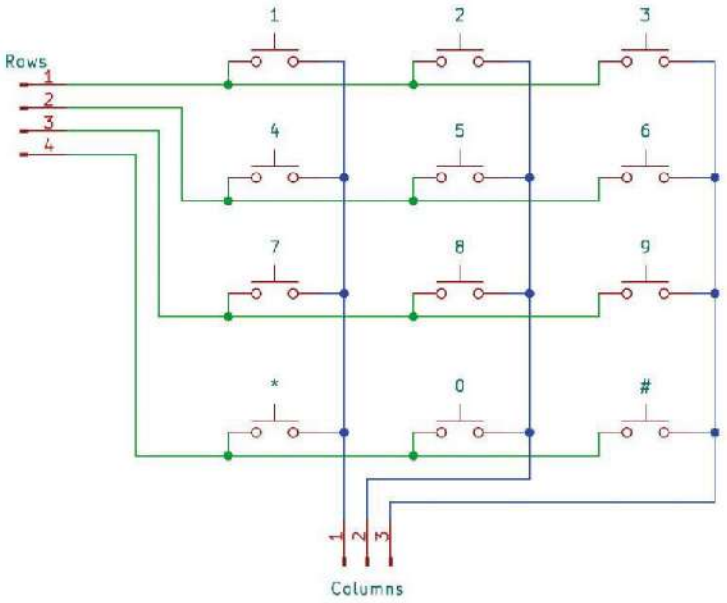


Figure 3 An alternative is to use a matrix keypad. These have the buttons arranged in a grid of rows and columns, which reduces the number of pins needed

08 Reading the keypad

The keypad is read by using the `get_num_pads` method. This returns a value between 0 and 65535 depending upon which buttons are pressed. There are 16 keys, but we've only used the first eight to simplify the explanation. **Figure 5** shows an overview of how the binary number system works in relation to the keypad numbers.

Inside the microcontroller, the status of the keys is represented by a binary digit for each key. If the key is pressed, then it has a value of 1 and, if it is not pressed, then it has a value of 0. The rightmost digit is key 0 which, if pressed, has a decimal value 1. If button 1 is pressed, that has a value of 2. That goes up to 128 if button 7 is pressed.

09 Shifting the values

To determine which button is pressed, you can read each of the bits in turn and see which are set to 1.

Starting at the right, look at the rightmost digit which is a 0, which means that button 0 is not pressed. Shift the numbers to the right and the rightmost digit becomes a 1, which means that button 1 is pressed. In Python, the shift is done using >> (two greater-than characters) followed by the number of bits to shift.

To perform the comparison, the bitwise AND function is performed. If you compare using '& 0x01' that will ignore all but the rightmost bit.

10 Creating a game

To turn this into a game, a random button is chosen which is then lit up. The player must try and press that button before the timer runs out. If the player presses the button before `lit_duration` is passed, then they score a point. If they don't, then the button turns red and the player loses

Keypad number	7	6	5	4	3	2	1	0
	128	64	32	16	8	4	2	1
Binary digit	1	0	0	0	0	0	1	0

Figure 5 An example of the binary code showing keypad numbers 1 and 7 pressed. The numbers above the binary digits are the decimal equivalent for each key

Figure 4 The TCA9555 port expander is the small square integrated circuit mounted on the rear of the keypad PCB

a life. After all lives are lost, the grid lights up green to show the final score, with each button representing four points.

11 Adding a challenge

To make the game more challenging, each time a point is scored, the time for the next button is reduced by 30 milliseconds. Whenever a life is lost, it is increased by 500 ms to give a reasonable chance to score more points. This makes it possible to get a reasonable score, with the difficulty increasing during the game.

Raspberry Pi Pico does not have a real-time clock, so to work out the time that has elapsed, `time.ticks_ms` is used. This measures time in milliseconds. The method `ticks_diff` is used to test whether the `lit_duration` time has been exceeded.

12 Further development

The game has some areas for improvement. Firstly, it will only run once and then you need to restart it from Thonny. Another `while` loop can be added to keep the game running, along with resetting the score and other variables.

Whilst it is possible to get a range of scores, many players will score around the same amount. This is due to the way that the complexity increases linearly. An alternative is to increase the difficulty using levels instead.

Another improvement would be to have multiple buttons light at the same time. See version two in the GitHub repository for an improved version of the game.

mole-bop.py

DOWNLOAD THE FULL CODE: magpi.cc/molebop

Language: **MicroPython**

```
001.  import time
002.  import random
003.  import picokeypad
004.
005.  keypad = picokeypad.PicoKeypad()
006.  keypad.set_brightness(0.5)
007.  NUM_KEYS = keypad.get_num_pads()
008.
009.  lit_button = random.randint(0, NUM_KEYS - 1)
010.  lit_time = time.ticks_ms()
011.  lit_duration = 1000
012.
013.  score = 0
014.  lives = 3
015.
016.  def all_off():
017.      for i in range (0, NUM_KEYS):
018.          keypad.illuminate(i, 0,0,0)
019.
020.  while True:
021.      # Check to see if the time has expired
022.      now_time = time.ticks_ms()
023.      if (time.ticks_diff (now_time, lit_time) > lit_duration):
024.          # light it red for missed
025.          keypad.illuminate(lit_button, 255, 0, 0)
026.          keypad.update()
027.          time.sleep (1)
028.          lives -= 1
029.          if (lives <=0):
030.              break
031.          # make the next one a little easier by adding 400ms
032.          lit_duration += 500
033.          # choose new button
034.          lit_button = random.randint(0, NUM_KEYS - 1)
035.          lit_time = time.ticks_ms()
036.
037.      # Turn LEDs off and the lit one on
038.      all_off()
039.      keypad.illuminate(lit_button, 255, 255, 255)
040.      keypad.update()
041.
042.      # scan keys to see if the lit_button key is pressed
043.      button_states = keypad.get_button_states()
044.      for i in range (0, NUM_KEYS):
045.          if i == lit_button and button_states & 0x01 > 0:
046.              score += 1
047.              lit_duration -= 30
048.              lit_button = random.randint(0, NUM_KEYS - 1)
049.              lit_time = time.ticks_ms()
050.          button_states = button_states >> 1
051.
052.  print ("Score {}".format(score))
053.  all_off()
054.  if (score > 64):
055.      score = 64
056.  for i in range (0, score / 4):
057.      keypad.illuminate(i, 0, 255, 0)
058.  keypad.update()
```

Part 03

Camera Module: More on video capture

Discover in-depth settings for capturing video with Raspberry Pi Camera Module

MAKER

David Plowman

David is an engineer at Raspberry Pi with a special interest in camera software and algorithms, and image processing hardware.

raspberrypi.com

You'll Need

- Raspberry Pi
- Camera Module **magpi.cc/cameramodule**
- Raspberry Pi OS

We met libcamera-vid at the end of last month's Camera Module tutorial (magpi.cc/130), and found out how to record a short video clip. Like libcamera-still, libcamera-vid has many more options for controlling the resolution, frame rate, and other aspects of the video that we capture, and we'll discover some of those in this tutorial.

Changing the video resolution and frame rate

By default, libcamera-vid will capture videos at a resolution of 640×480 pixels. But it's easy to change this with the `--width` and `--height` parameters (just like libcamera-still). For example, if you want to capture a video that is 1920×1080 pixels (known as 'full HD') instead, please use:

```
libcamera-vid --width 1920 --height 1080 -o
full_hd.h264
```

This particular resolution of 1920×1080 pixels should be regarded as the largest resolution that can be captured in video, unless noted otherwise in the text below.

Because we're talking about videos now, we can also ask for different framerates. By default, libcamera-vid will deliver 30 frames per second (fps). We can ask for a different frame rate using the `--framerate` parameter:

```
libcamera-vid --width 1280 --height 720
--framerate 50 -o 50fps.h264
```

This will record a 720p video at 50 fps, so long as the camera can deliver the frames that quickly (otherwise you will get the maximum frame rate that the camera can supply).

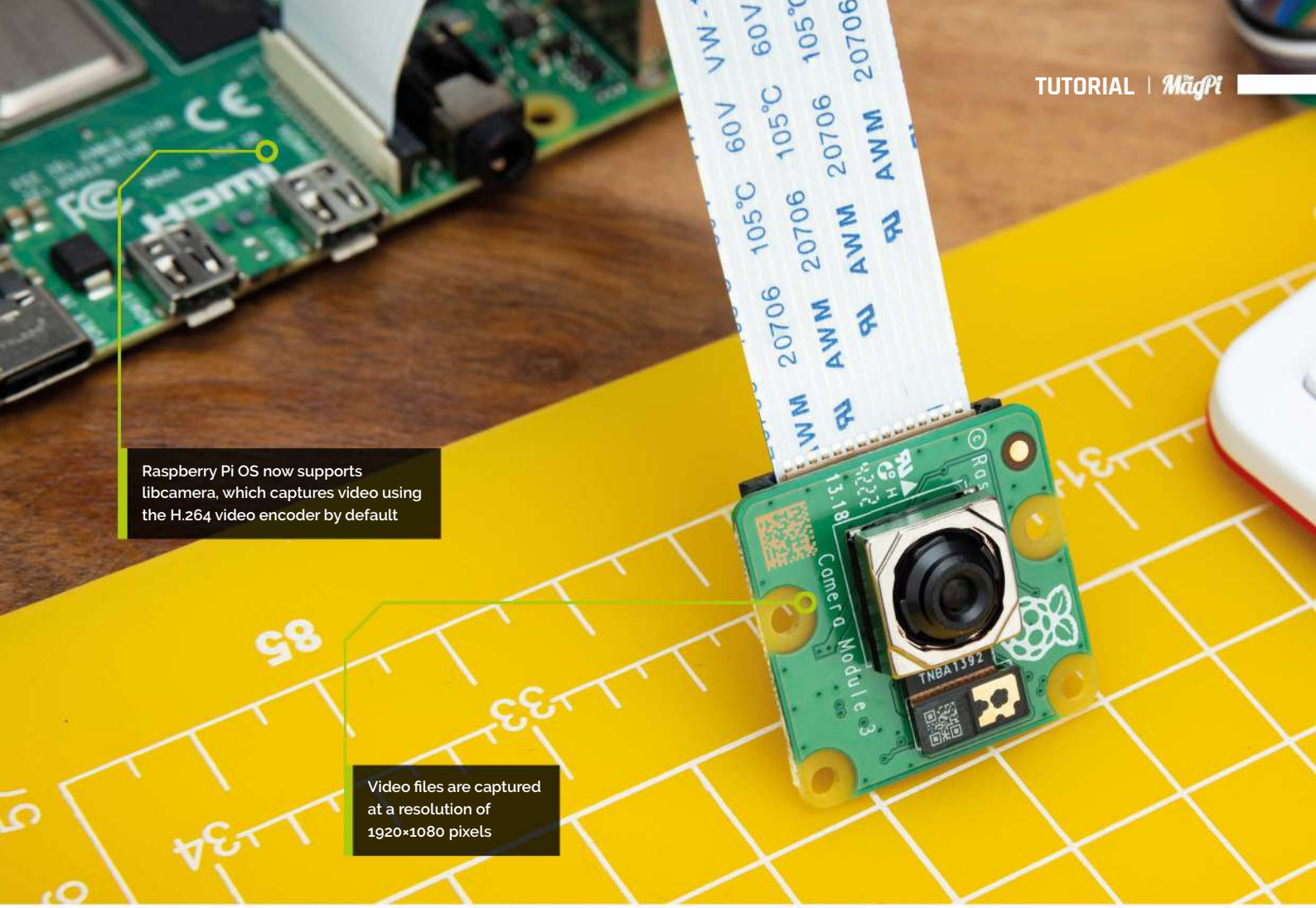
Video encoders and file formats

Videos, just like still images, are normally compressed before saving them to disk. In fact, it's even more important to compress video frames because, with hundreds or thousands of frames, the output file will explode in size.

After compression, the video frames are written to a file with a specific 'container format'. There's a choice as to how to organise video frames in a file, not least because there can be other data – such as audio – in the file too, and the container format defines exactly how playback software can access and decode the information.

H.264 Files

The default encoder used by libcamera-vid is the H.264 video encoder. There is special hardware to support this so it combines both


Raspberry Pi OS now supports libcamera, which captures video using the H.264 video encoder by default
Video files are captured at a resolution of 1920×1080 pixels

good performance and good compression. Once compressed, the frames are written back to back directly to a file which we normally describe as an 'H.264' file, and by convention often denote it with the file extension '.h264'.

It's worth noting that these H.264 files are very simple; they're not really true container formats. That is, you couldn't mix audio data in with the video, and the video frames do not even have timing information associated with them. Support for playing back such files can be limited, and even where it is supported, the software will have to guess a framerate for the video.

Popular media player software VLC used to support the H.264 file format, but more recently no longer plays them correctly. For this reason, we recommend other programs, such as FFplay.

We've already seen how to record a video with a different resolution; another useful option is the bitrate parameter (`--bitrate` or just `-b`). This controls the size of the video file created, and therefore the amount of compression and the perceived quality. Compare the following:

```
libcamera-vid -b 1000000 --width 1920
--height 1080 -o bad.h264
```

...versus:

```
libcamera-vid -b 9000000 --width 1920
--height 1080 -o good.h264
```

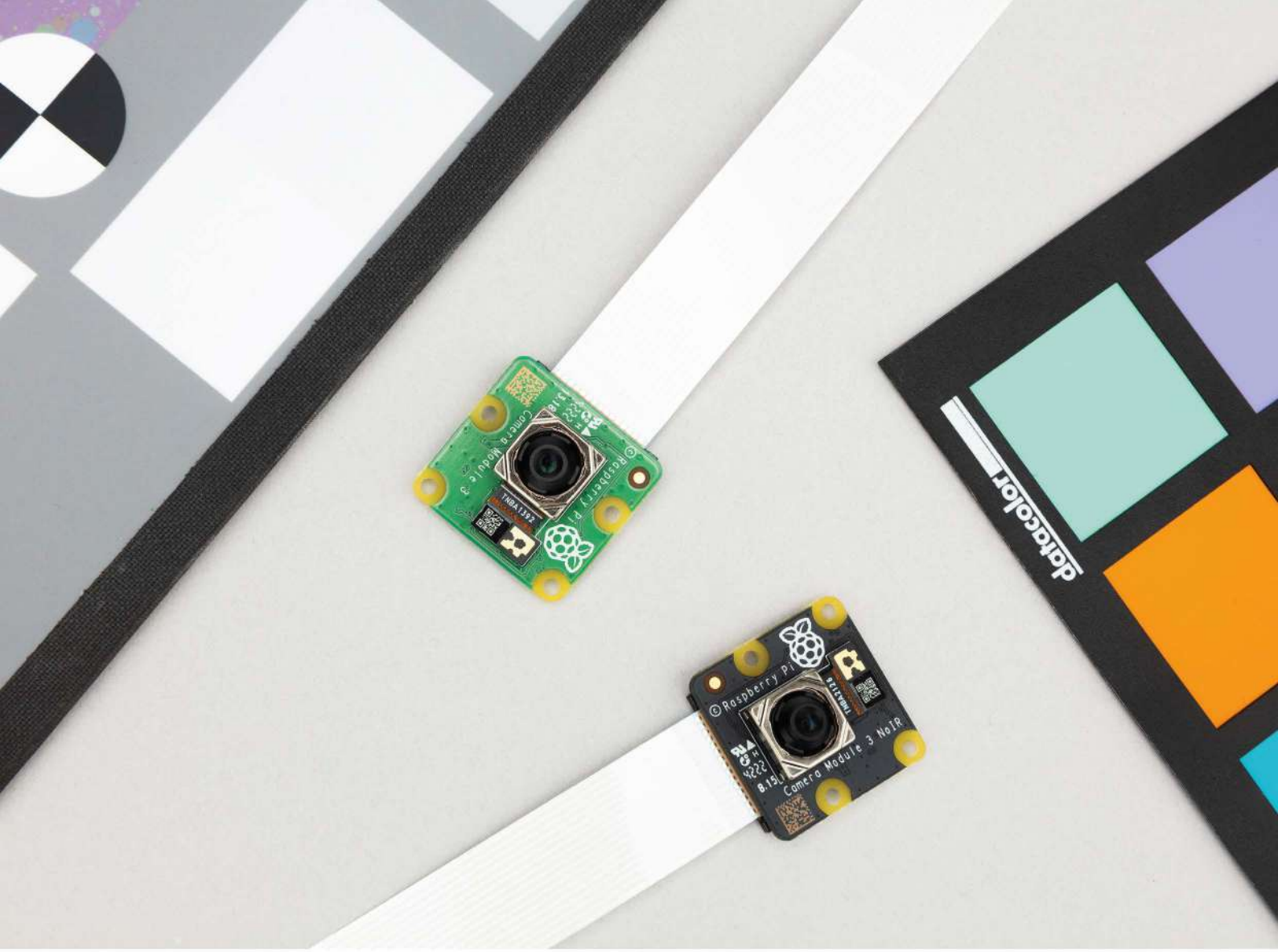
The 1,000,000 bits per second (1 megabit per second, or 1Mb/s) video ('bad.h264' option) shows far more objectionable compression artefacts than the 9Mb/s 'good' video.

> There's nothing to stop you from compressing one still image after another (very quickly)

MJPEG files

We've previously encountered the JPEG image encoder for compressing still images. There's nothing to stop you from compressing one still image after another (very quickly) and saving them back to back to create a video file. That's exactly what an MJPEG file is.

There are some drawbacks, however. The MJPEG file format grew up organically and so lacks some clear decisions on exactly how to format the JPEGs within them. Often they work well, but support can be a bit patchy.



▲ Video files are compressed to save vital storage space

To create and save a 20-second MJPEG file, use this:

```
libcamera-vid -t 20000 --codec mjpeg -o
test.mjpeg
```

It is necessary to specify the `--codec` option to get a proper MJPEG file – changing the file name on its own is not enough. Using the same JPEG encoder as libcamera-still, it supports the same JPEG quality parameter (`--quality` or `-q`), though in video files we can normally get by with lower values – indeed the default is only 50. An MJPEG file will ignore any setting made by `--bitrate` (or `-b`).

You can play an MJPEG file with ffplay:

```
ffplay test.mjpeg
```

MJPEG files, in contrast to H.264 files, can be recorded at resolutions greater than 1920×1080 pixels, though increasing the resolution will decrease the framerate at which libcamera-vid can keep up, resulting in dropped frames and possibly choppy videos.

Uncompressed video files

Most video compression formats are 'lossy', meaning that the video you save will have some degree of quality loss compared to the original camera images. Therefore, it's also possible to save completely uncompressed video, with the caveat that these files can become extremely large, and you would need to know exactly how you intend to use them.

To capture an uncompressed video use the `--codec yuv420` option as follows:

```
libcamera-vid --width 320 --height 240
--codec yuv420 -o test.yuv420
```

The relatively small size here (320×240 pixels) means the file size won't explode too rapidly, and also that the system won't get bogged down trying to write vast amounts of data to the disk (which can easily become quite slow).

Remember that, as usual, the `--codec` option is necessary and that changing the file name on its own is insufficient.

Finally, the format of the output file is a simple dump of each of the uncompressed image frames,

one after the other. These are in 'YUV420 planar' format, which you will need to understand in order to make use of it. The use of these files lies beyond the scope of this guide, and is not recommended for beginners.

Uncompressed video files also have no resolution limit (as H.264 files do), though large resolutions will very quickly become a bottleneck due to the volume of data being written out.

> The format of the output file is a simple dump of each of the uncompressed image frames

MP4 files, audio and other container formats

We've seen that H.264 and MJPEG video files are not without certain limitations. The video file format with probably the best support is the MP4 file, and libcamera-vid can create these directly, even mixing in an audio stream if you have a microphone.

To create an MP4 file, we have to choose a different codec. It will still use Raspberry Pi's hardware H.264 encoder, but will access it via a third-party library that will take care of the MP4 container.

To record an MP4 file without audio, use:

```
libcamera-vid --codec libav -o test.mp4
```

The usual options that we've seen so far (`--timeout`, `--width`, `--height`, `--bitrate`) will all work as before, though the other options discussed later in this tutorial won't (unless stated otherwise).

To create an MP4 file with an audio stream, use:

```
libcamera-vid --codec libav --libav-audio -o
test.mp4
```

This will make certain reasonable default choices about the audio and how to encode it; please refer to our online documentation for further details.

The 'libav' encoder can handle other container formats such as MKV files or MPEG2 Transport Streams. It is possible to stream the latter directly over a network. Please refer to our online documentation for more information (**magpi.cc/cameradocs**).

Pausing and resuming recordings

You can, of course, pause and resume a video recording by stopping libcamera-vid (for example with **CTRL+C**) and restarting it again when you want the recording to resume. But you may recall how we could get libcamera-still to capture images when we press a key, and we can get libcamera-vid to pause or resume a recording in just the same way.

There are a few more options that we need to cover:

- **`-t`** or **`--timeout`** The length of time, in milliseconds, for which libcamera-vid will run. This may often be zero if you're going to press a key to terminate the application.
- **`-k`** or **`--keypress`** Pressing the **ENTER** key toggles between recording and not recording. Type 'x' and press **ENTER** to quit.
- **`--initial`** Followed by pause or record. This starts libcamera-vid in the paused (i.e. not recording) or recording state. Pressing the **ENTER** key will then start or pause the recording, respectively.

Raspberry Pi Camera Module 3 features a 12MP sensor, autofocus, and HDR support

Top Tip

Save space

After recording finishes, the compressed file is saved to disk. Use the Circular Buffer option to save a specific number of seconds from the end of the capture.

- **`--split`** Every time a recording is resumed, the recording starts writing a new output file.
- **`--inline`** Tells libcamera-vid to insert certain extra header information at the start of every file that it writes. You should normally include this when using the `--split` option.
- **`-o`** or **`--output`** The name of the output file or files. When we're using the `--split` option, we'll often want each output file to have a different name, so we can use the same 'counter syntax' that we saw with libcamera-still's timelapse option. For example, `-o test_%03d.h264` will write the output files **test_000.h264**, **test_001.h264**, **test_002.h264** and so on.

Let's try an example:

```
libcamera-vid -t 0 --inline --split
--initial pause -k --width 1280 --height 720
-o video_%03d.h264
```

This will start libcamera-vid in the paused state, so it won't be recording anything yet. When **ENTER** is pressed, it will start recording **video_000.h264**. It will pause when **ENTER** is pressed again, and then start recording **video_001.h264** when **ENTER** is pressed for the third time. This will carry on indefinitely until 'x' is typed followed by **ENTER**.

Recording in response to a signal

For those familiar with Linux signals, an alternative to pressing a key is to send a signal instead. To do this, simply use `-s` or `--signal` instead of `-k` (or `--keypress`).

To send a record/pause signal to libcamera-vid, first start it and then type the following into another terminal window:

```
kill -SIGUSR1 `pidof libcamera-vid`
```

And you can force libcamera-vid to quit with:

```
kill -SIGUSR2 `pidof libcamera-vid`
```

This should be familiar as it matches libcamera-still's behaviour!

Other recording options

libcamera-vid has many other parameters and we'll look very briefly through a few of them here.

Circular buffer output

One useful feature is the ability to keep writing the recorded video stream to a buffer in Raspberry Pi's memory. This is a limited size, and when it fills up, the earliest data in the buffer is evicted to make space for the new frames. When libcamera-vid exits, this circular memory buffer is flushed to the disk. This enables you to leave the camera running indefinitely, but to save only the last several seconds of data when some event occurs that causes the program to be stopped.

Let's see an example:

```
libcamera-vid -t 0 --keypress --circular
8 --inline --width 1280 --height 720 -o
circular.h264
```

This will run indefinitely writing to a circular memory buffer, and when the user types 'x' and **ENTER**, the program will quit after saving the last 8MB (megabytes) of data to the file **circular.h264**. The size of this memory buffer is determined by the number (which is in units of megabytes) after the `--circular` option.

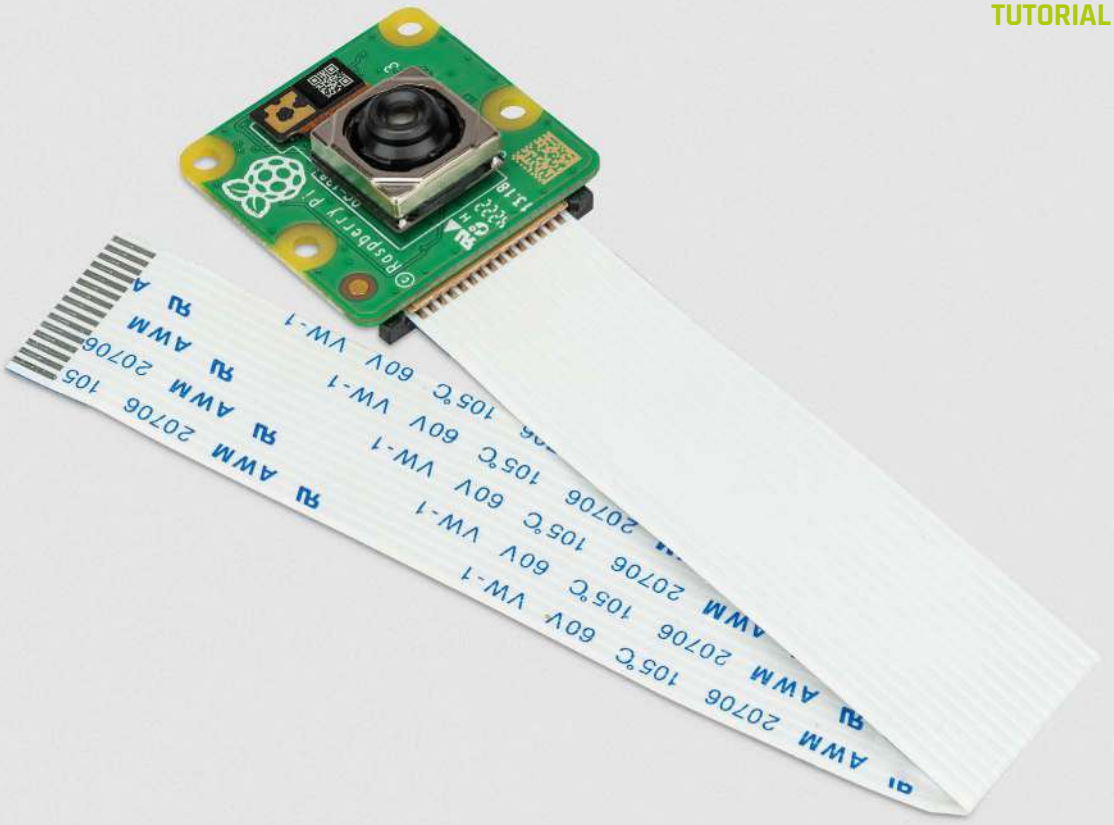
You can use `--signal` instead of `--keypress` if you prefer.

> For those familiar with Linux signals, an alternative to pressing a key is to send a signal instead

Timing Information

We remarked earlier how H.264 format files contain no timing information. libcamera-vid allows this information to be output to a separate file for later use or analysis.

Here we simply specify the `--save-pts` option followed by a file name where the frame times



When using Camera Module 3, autofocus is enabled automatically in continuous mode

from the start of the video are stored in text form in units of milliseconds. For example:

```
libcamera-vid --save-pts timestamps.txt -o
circular.h264
```

The file **timestamps.txt** will (for a 30 frames per second recording) start like this:

```
# timecode format v2
0.000
33.332
66.664
99.996
```

...and so on.

Autofocus and high dynamic range

Autofocus and high dynamic range imaging are supported at the time of writing only on Raspberry Pi Camera Module 3. The support mirrors that in libcamera-still very closely.

Autofocus

When using Camera Module 3, autofocus is enabled automatically in continuous mode. This means that the camera lens will move whenever necessary to maintain optimal focus on the centre part of the image, and this is probably what most users will want most of the time.

It's also possible to turn off autofocus and set the focus position of the lens manually. To do this, use the `--lens-position` parameter, and pass it a value measured in 'dioptres', meaning the reciprocal of the focus distance. Thus, to focus at a very close distance of about 0.1m, pass in the value 10 (which is 1 / 0.1). To focus at infinity, pass in 0 (informally, the value of 1 / infinity). You can pass in non-integer values too. For example:

```
libcamera-vid --lens-position 0.5 -o 2_
metres.h264
```

...will set the focus position to two metres and not move the lens again.

High dynamic range imaging

The Camera Module 3 supports high dynamic range (HDR) imaging. To use it, specify the `--hdr` option on the command line, for example

```
libcamera-vid --hdr -o hdr.h264
```

Note that non-HDR captures can be performed at a maximum resolution of 4608×2592 pixels, but HDR captures, because of the special nature of the sensor required to support HDR, are limited to 2304×1296 pixels (exactly half the width and height of the non-HDR mode). H.264 video files are, in any case, limited to 1920×1080 pixels, so this restriction is often not relevant.

Rad your ride with Pico

Turn a manual scooter into a motorised monster

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Scooters are fun, and can be a good alternative to cars and bikes when considering the daily commute. They can even be useful while at work if your job involves moving between campus buildings several times a day, or travelling back and forth in large warehouses or hangars. Electric scooters are even more fun, but they are also expensive. In this project, you'll see how to take an ordinary manual scooter and make it electric without resorting to expensive hub motors or conversion kits. This is a two-part project. In the first part, we'll take a look at the components we'll use and how we'll need to modify our scooter to make it all work. In the second part of the project, you'll see how to power your motor from custom batteries.

The anatomy and function of an electric scooter isn't mysterious or difficult to understand. The user stands on a platform, activates a throttle, a motor hums into life, and the wheels of the scooter begin to turn. When the user wants to stop, they release the throttle and apply a brake or use their foot to slow down.

A manual scooter works in a similar way, but with more leg action and less throttle – surely we just need to add a motor, some batteries, and a speed controller, and that's everything, right? Well, it isn't quite that simple. Firstly, manual scooters come with very primitive brakes (if they have brakes at all). The wheels on the scooter are typically not designed to be driven, and just have a simple bearing and possibly a fork with a suspension spring. Making a manual scooter move under motor power means making some very permanent changes to the chassis of the scooter, mounting the motor near to the rear wheel, enlarging the rear fork to accommodate better brakes and a drive sprocket on the rear wheel, and modifying the

wheel itself so that it can handle the extra stress coming from the motor.

Most of the issues with the wheel can be dealt with by using some 3D-printed parts and a healthy dose of epoxy resin to reinforce the wheel. The drive sprocket for the motor and the disc to upgrade the brakes can be mounted on opposite sides of the rear wheel using 3D-printed parts, and using plenty of resin to fill the space between the two sides will make sure that the bolts have plenty of solid wheel material to make contact with when the forces from the motor and brakes are transferred through them.

> **Once you have a solid wheel, you can build your** rear fork to match its width

Once you have a solid wheel, you can build your rear fork to match its width. If you have access to a welder, you can do this very quickly by cutting the original rear fork into pieces and re-welding it with →

OFF-ROAD

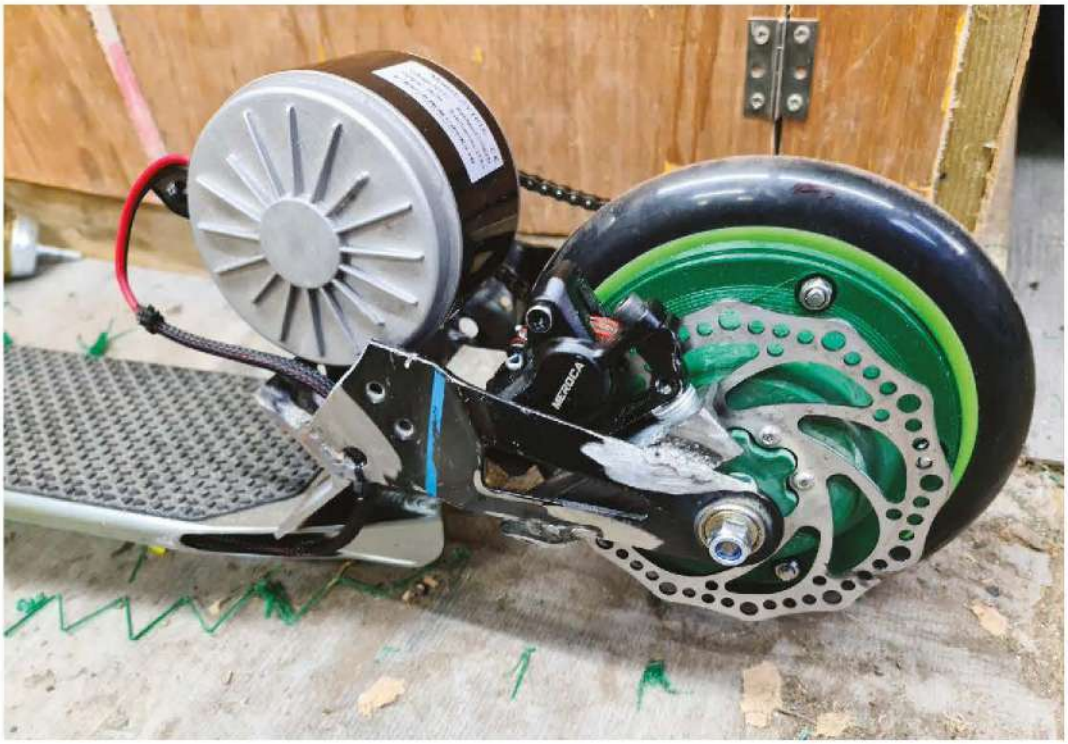
If you're thinking of using an electric scooter in the UK, you're going to be disappointed to learn that you're not going to be able to ride it on the open road, or on a footpath, or even in a bike lane. Electric scooters are in a class of vehicle known as 'powered transporters'. It's the same place you'll find one-wheels, hovercrafts, and electric skateboards. In UK law, a powered transporter is the same as a motor vehicle, and as such you would be expected to license and maintain it in exactly the same way as a car or motorbike. Unfortunately, scooters don't have some of the important features needed to pass an MOT (Ministry of Transport) test (like headlights, indicators, registration plates, etc.) – and so they will not be legal to ride on the road, and you can't get insurance to drive them. However, they are still technically classed as motor vehicles, so you aren't allowed to ride them on the pavement either. If you *do* ride on the pavement or on the road, you could be fined several hundred pounds and risk losing your driving licence. You can ride them on private land – such as corporate campuses – with the landowner's permission. At the time this article was written, the government is developing new standards and adjustments that will eventually make electronic scooters road-legal in the UK. How long that process will take depends on how quickly the government decides to move.

TAKE CARE

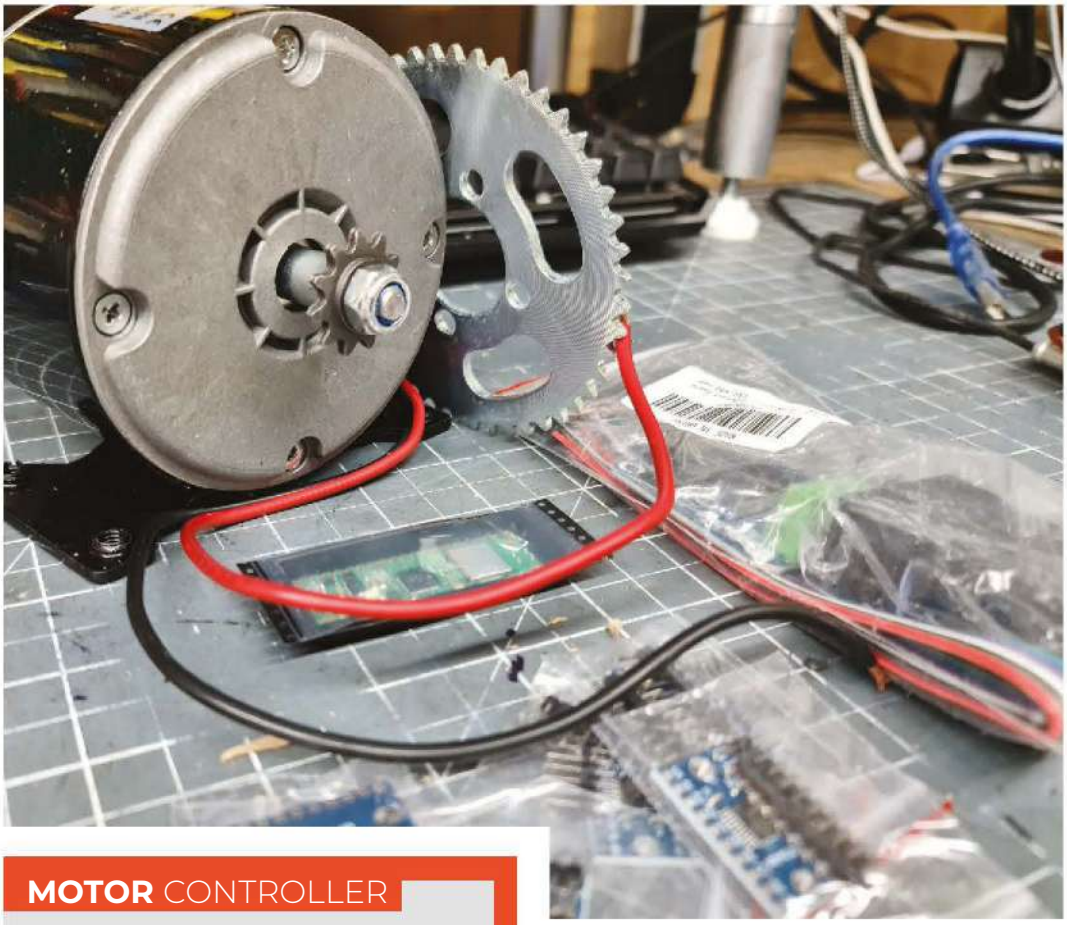
Scooters – whether powered or not – carry some risks. It's your responsibility to make sure the scooter doesn't go faster than you personally feel comfortable with. You are also responsible for ensuring that any chassis modifications you make are sufficiently strong. We strongly recommend you err on the side of caution as your decisions on both these risks could impact not just you, but the people around you.

QUICK TIP

Cycle chains can be joined with a split link, so you don't have to buy a chain exactly the right size. Simply remove excess links, join the chain with the split link, then tension the chain by moving the motor.



Left
Most scooters don't really bother with an effective brake, because they don't go very fast and you can stop them with your foot. If you're on a motorised scooter, you'll need a more powerful rear brake to arrest your more considerable forward momentum. A disc brake from a bicycle will do the job. Just remember that the brake will only stop the scooter, not you



Right
Using a motor with a sprocket and chain is much cheaper than using a hub motor and allows you to increase the torque from the motor mechanically

HackSpace

This tutorial is from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has amazing tutorials. Find out more at **hsmag.cc**.

QUICK TIP

In MicroPython, the Pi Pico PWM output ranges between 0 and 65025, so setting a PWM pin with `pwm.duty_u16(32512)` would give you about 50% power.

MOTOR CONTROLLER

Most BTS7960 motor controllers are H-bridge controllers that have a similar pinout. The boards have two power inputs, one of which (often labelled as pins B+ and B-) supplies power for the motor. Next to this input are the connections (M+ and M-) that directly attach to the motor.

The other power input is used to supply the logic side of the board, and is labelled as Vcc and GND. This logic level power input runs at 3.3 V or 5 V and is isolated from the motor power connection for safety. With a Raspberry Pi Pico, you should be powering this with 3.3 V. The remaining connections on the board are used to control the speed and direction of the motor, and optionally monitor the current used by the motor. We won't be using the current-sensing features, so the pins labelled R_IS and L_IS can be ignored. The L_EN and R_EN pins enable the motor to turn clockwise or anticlockwise when they are high, or disable the motor when low. Obviously, only one of these should be enabled at a time. The LPWM and RPWM pins accept a PWM signal from the Pico, or a solid logic high to power the motor at 100% power. Since a scooter generally doesn't need a reverse gear, you only need to connect either the left or right enable and PWM pins to the Pico, depending on which way around your motor is connected. Assuming that you have a right-hand motor setup, you would make the motor move by setting the R_EN pin HIGH and providing a PWM signal to RPWM.

> **You will probably find that you won't want to run your** motor at 100% speed under any circumstances

a larger gap. If you don't have a welder, you can achieve similar results using threaded bar and locking nuts with a spacer between each side of the fork.

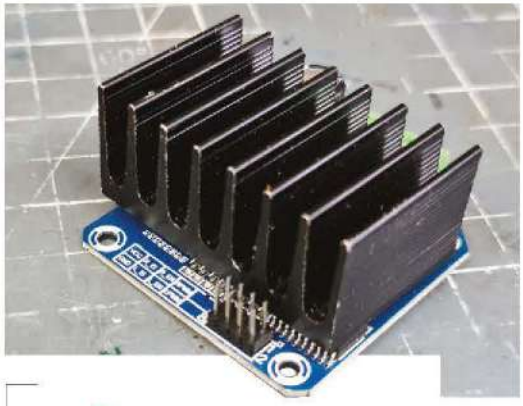
For something like a light scooter, a 20 amp, 24 volt motor in combination with the gear reduction from the sprocket drive should have plenty of power to move an adult at terrifying speeds. By controlling the power to the motor using PWM signals from a Raspberry Pi Pico, you can provide very smooth speed control without losing as much torque as you would if you simply reduced the overall supply voltage. You will probably find that you won't want to run your motor at 100% speed under any circumstances, and will throttle the maximum PWM value to something less than bowel-voidingly high speeds.

ANALOGUE IN

Setting the motor speed is only half of the process; you also need to get a value from the user by reading a throttle control. It's tempting to think about creating a simple throttle control using a potentiometer, but that would be a bad idea. If you've ever used an old radio, you'll know that the potentiometer used in the volume control gets worn out, and the volume of the radio will suddenly change or crackle at full volume for a fraction of a second. This is annoying in a radio, but if it happens on a motor controller and the motor suddenly speeds up to 100% without warning, the result could be catastrophic. To avoid this problem, most electronic throttle controls use a Hall probe and a magnet to ensure smooth speed changes. The Hall probe and magnet aren't in physical contact and there are no tracks to get worn, dirty, or broken. The probe normally has two power pins and a signal pin, which outputs an analogue value to reflect the strength of the magnetic field it detects. As the magnet moves closer, the field gets stronger. Checking the position of the throttle controller is as simple as connecting up and polling the analogue pin of a microcontroller, just as you would with a potentiometer.

Above
You will need to mount your motor so that the motor's sprocket matches up with the sprocket on the wheel. The easiest way to do this is to fix a mount for the motor on the rear of the fork with slotted holes, and use the adjustment that the slots provide to position the motor correctly. You can also use this technique to tension the chain between the motor and the wheel

Left
You can see here how much wider the wheel becomes once you allow for the sprocket and brake disc. Widening that rear fork might seem a bit daunting, but it's actually pretty easy to do. Getting rid of the in-built suspension makes things much easier if you're struggling, and it won't actually make that much difference to the feel of the scooter



Above
To control a large motor, you will need to use a powerful motor controller like the BTS7960, which is often found on motor control modules that can handle around 43 amps, when appropriately cooled

Interactive stream lights

Have your viewers change your lighting during streams with this Raspberry Pi Pico-powered setup for major streaming services

MAKER

Rob Zwetsloot

Rob knows a hedgehog that streams model kit building, which needs decent lighting for viewers to see what Gundams are being made.

magpi.cc

Interactive lights are hardly a new thing for streamers. If you have certain smart lights, you can connect them to Lumia Stream (**lumiastream.com**) and go from there. Of course, that's not as fun as making it yourself, and with a Raspberry Pi Pico W and some NeoPixels, you can get one set up pretty quickly.

We've borrowed some code and ideas from Sean O'Steen's excellent Raspberry Pi Pico Beret (**magpi.cc/picoberet**) which has a web server and animation controller running on different cores on Pico. Now, let there be light.

You'll Need

- Raspberry Pi Pico W **magpi.cc/picow**
- NeoPixel LEDs **magpi.cc/neopixels**
- Stream bot, such as Mix It Up **mixitupapp.com**

01 Ready your LEDs

For this use case, we just need to illuminate the area of a cutting mat – a relatively very small area compared to a room. As well as helping with video quality, the benefit of these lights is that it will also help the streamer see what they're doing. You don't want to blind yourselves with the LEDs!

We have a small strip of eight NeoPixels that we've cut off a reel, which will provide plenty of light to the desk when positioned in the correct spot (see Step 7).

02 Wire it up

Refer to **Figure 1**'s Fritzing diagram on how to wire up the NeoPixel strip to Raspberry Pi Pico W. It's fairly simply though – Pico can provide power straight to the NeoPixel using the 5V pin, as well as providing a ground for the circuit. Data is sent via a single pin which tells each LED how to act.

For larger lengths of NeoPixels, a 220 Ohm resistor between the 5V pin and the NeoPixel strip is best practice to stop any of the LEDs from overheating.

Make sure your Pico is in range of a micro USB cable plugged into a PC so you can program it too.

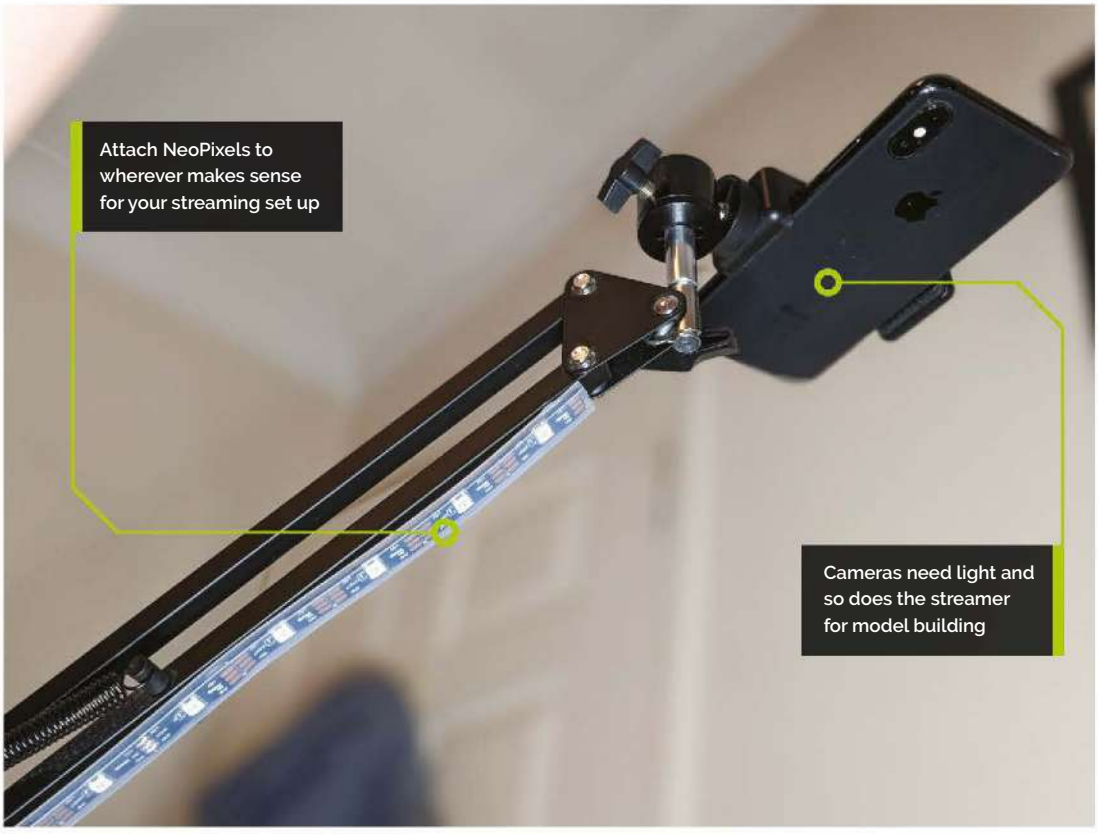
03 Getting the code

We've uploaded the code for this to our GitHub, and you can find it at: **magpi.cc/streamlights**. It comes in two parts – **stream_lights.py** and **secrets.py**. You'll need to modify both for your specific setup; **secrets.py** just need to have your wireless network info added, while **stream_lights.py** code will need to be updated with your number of LEDs on line 15, **NUM_LEDs**.

04 Code explained

The reason a Pico W is needed for this project is that we're making use of its web server capabilities. The idea is that once booted, Pico W will run the code and set up the web server that can be accessed via your local network. Sending web requests in the form of URLs is captured by the server and when a specific request is processed, it will activate a different light sequence.

Both parts of this code run on their own threads so they're able to work independently of each


Attach NeoPixels to wherever makes sense for your streaming set up
Cameras need light and so does the streamer for model building

other – RP2040 has two cores making this much easier to do.

The light sequences are done in a traditional NeoPixel manner of setting the LEDs with simple RGB values.

05 Set up Pico

Get Thonny on your PC, or boot up your nearest Raspberry Pi. Hold down the BOOTSEL button, and connect Pico to your computer via a USB cable. Once that set up is complete, open Thonny, and click the Python version on the bottom-right of the window. There you'll get the option to install MicroPython to Pico, and in the following window you'll need to select firmware for Pico W/WH.

> "Testing at every step is very important"

Once that's done you'll need to save the **stream_lights.py** file to Pico from Thonny and rename it as **main.py**. Save the **secrets.py** file to Pico as well. Restart Pico and it will give you the IP address in Thonny you'll need to send web requests.

06 Test it out

Testing at every step is very important. So, before we go any further, turn your Pico on and wait for the lights to turn on. Once that's done, open a browser on your computer on the same local network, and enter the IP address from the previous step along with the URL for a specific light command. For purple in this example, you'll need to use something like: 192.168.1.100/?purple.

Check out some of the other links to make sure they work as planned, or debug them if needed.

07 Secure your LEDs

For this use case, we're using a boom arm attached to a desk to hold a smartphone to use as a camera. With the arm going over the area we're capturing, we've attached the LED strip to the underside of the arm with some cable ties and they light up the area just fine. You could easily design and 3D-print your own stand, and we do

Top Tip

Bright white?

We've used white as the colour for the lights in the code example. However, that might not be the right tone for your stream. Try a warmer orange for a more relaxing look.

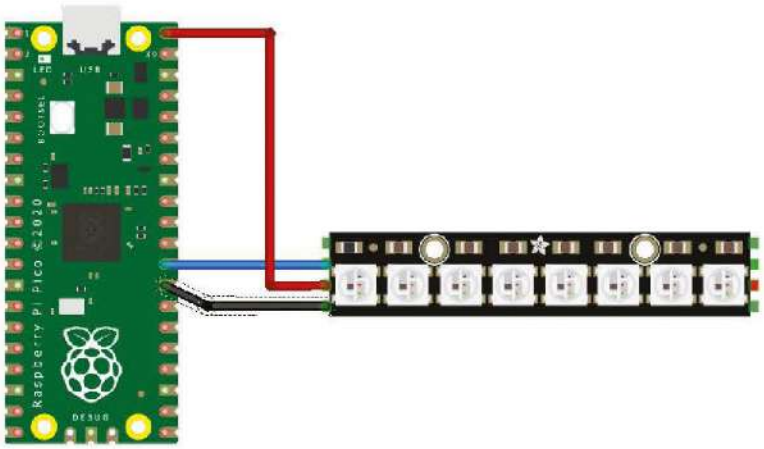


Figure 1: Pin 40 (top-right) is the 5v pin, while every square pin on Pico is ground

recommend getting some frosted acrylic to cover the LEDs so that the light on the desk area isn't too sharp.

Make sure a power supply can reach your Pico as well.

Top Tip

Lotsa lights

This example uses a small strip – for larger setups, look into using external power for the NeoPixels.

08 Set up Mix It Up

Mix It Up is a customisable stream bot that allows you to create event triggers for a stream – e.g., when a new person follows you, a video can be played on stream via an overlay in the broadcast software. Head to **mixitupapp.com** and download the installer to your streaming PC, install it, and connect to your streaming account (in our case, Twitch), and go through the intro screens.

Mix It Up is also a good companion to the custom stream deck we made in **magpi.cc/111**, as it is able to capture global hot keys sent by Keybow 2040 more consistently than some other pieces of software, and can manually activate the desired function.

Using Action Groups means you don't have to write out the web URL every time

09 Action groups

Creating a command on Mix It Up can be done directly on the trigger (such as a new Twitch subscriber) or you can create a universal command to be called for any trigger. Either way, the flow of the command is the same, and it's quite simple here.

Go to the menu in the top-left and select Action Groups. Click 'New Action Group' to open the command editor. From the Action list on the bottom of the window, select Web Request and press the add button. Enter the URL for the specific light animation you want to use, e.g., 192.168.1.100/purple, and then click save. From here you can test the command using the play button when Pico is running. We suggest toggling the Unlock Command option as well, which will allow other commands to run if an issue occurs with this one.

10 Text commands

There are two major ways for a chat to control your lights – one of the simpler ways is by using chat commands. From the menu, go to

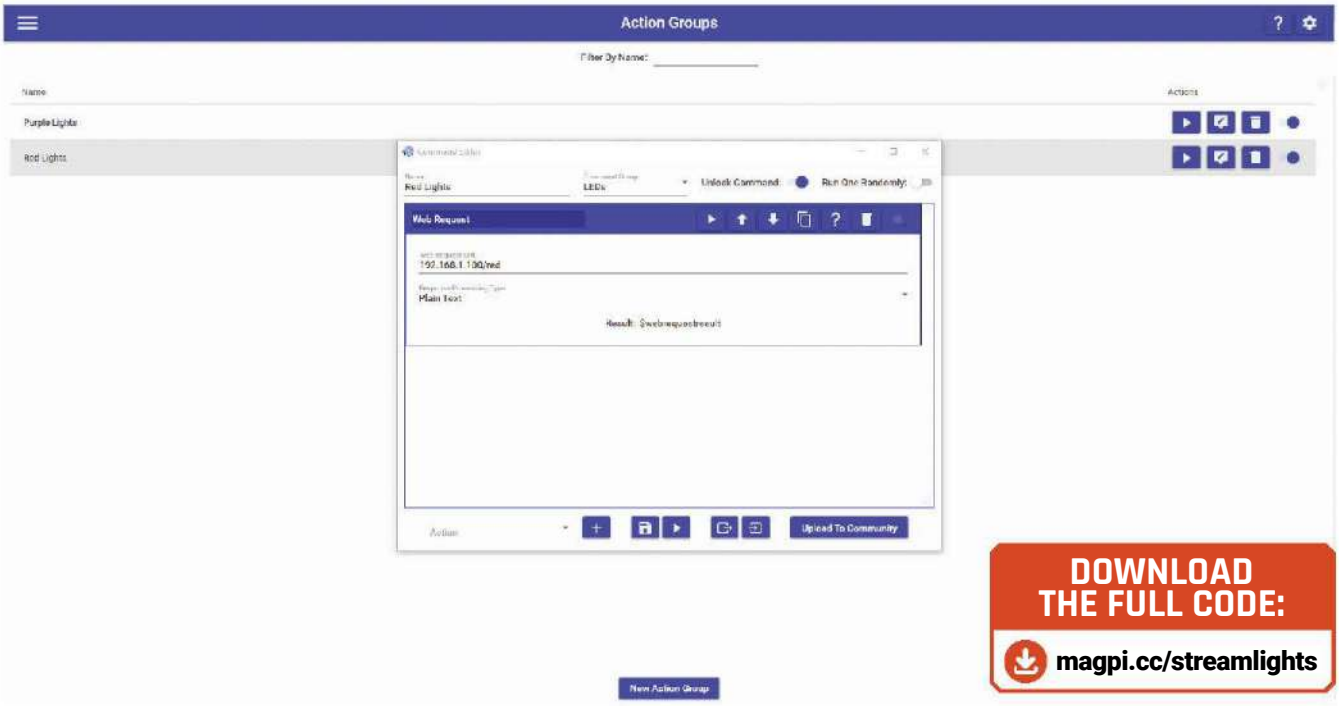
> "Make sure a power supply can reach your Pico as well"

Commands and then to the Custom Commands tab. We've made the chat trigger purplelight so that when someone in chat writes !purplelight (with the bang at the start) it gets activated.

From the Actions menu, choose Command, change the Command Type to Action Group, and select the appropriate action. From here, you can also add multiple actions, with pauses in-between, allowing you to change the light sequence without changing the code on Pico.

11 Twitch redeems

If you're a Twitch affiliate or partner, your viewers can collect channel points simply by watching your streams. These points can then be spent on redeems and are popularly used to play a sound or get the streamer to drink some water

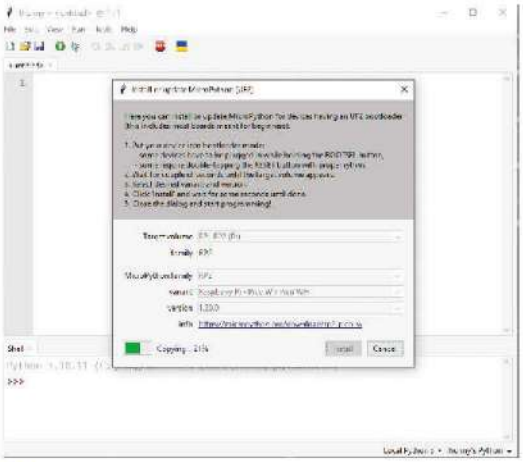


DOWNLOAD THE FULL CODE:
magpi.cc/streamlights

▲ Remember to use the IP address of your Pico W for the web requests

or stretch. Mix It Up is able to read the list of redeems and use them as triggers.

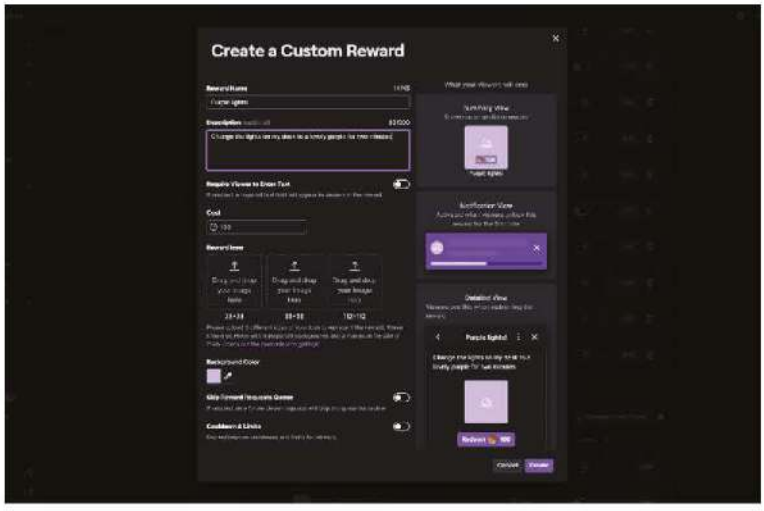
Go to the Twitch Dashboard, Viewer Rewards, and then Rewards and create a custom reward. Give it a name, a channel point cost, and consider giving it a cool-down so it can't be used continuously too. Once it's set up go to Twitch Channel Points in Mix It Up, Add Command to Existing Reward, and add the action group, like in step 10.



▲ Select the correct variant, as we require the wireless capabilities of Pico W

12 Happy streaming

Make sure to thoroughly test the actions before hitting live, and make sure to let your chat know about the new functionality of your stream! Whether it's showing off a Gundam or your wonderful visage, you may need to tweak the code to get everything looking just right. RGB codes for different colours can be quickly Google'd and Pico's code can be updated on the fly too. GLHF!



▼ Create a Custom Reward on your Twitch Dashboard

Raspberry Pi Camera Traps

Get photos – and video! - of your local critters.

By indoorsman **Rob Zwetsloot**

MONOFILM A600 300 MONOFILM A600

Raspberry Pi Camera Modules are wonderful things, and we've used them for countless projects in the pages of *The MagPi* over the years. You're coding them directly, so they're incredibly customisable, even being able to talk to the web or the multitude of sensors available for Raspberry Pi.

With all this power and a little bit of weatherproofing, it's incredibly easy to use a Raspberry Pi to take a look outside and find out about the natural world around you. Here are some projects to get you started...

Urchin's Gallery

The finest photos of the finest wildlife taken by the finest microcomputer

MONOFILM A600 800 MONOFILM A600

Rob unabashedly loves hedgehogs thanks to a blue one with attitude, and as it's his feature, he's decided to have a page dedicated to these ancient, spiky, and cute garden dwellers.

Photo by The Jolly Geographer and Wildlife Trust BCN

Photo by @adeibiza

- "Here's a front and back of Nora the hedgehog, taken with a My Naturewatch Raspberry Pi camera"
- "The hogs at the cat feeder, I 3D -printed some steps that they sometimes use"
- "Latest Hog Alley music video. More can be found @ hogcamb (mostly on Instagram). Not sure this kept me sane during lockdown but certainly busy."

Photo by Keith Freeman (@BiscuitKeith)

Photo trap projects

Build your own, cheap nature camera

By **Rob Zwetsloot**

Simple Raspberry Pi Camera Trap
magpi.cc/simplecamtrap

Housed in a Tupperware box as weatherproofing, this project makes use of an infrared camera with IR LEDs to light up the night. Not only that, but it teaches you how to set up MotionEyeOS to sense movement and take photos and/or video in the process – something you'll be able to customise to your own specifications if you plan to tweak the build.

Like My Naturewatch, it makes use of an air-tight food container, although we're not so sure about how weatherproof the camera is in that configuration. If you have an IR camera but not one with IR LEDs, you can actually get IR LEDs as individual components from most online electronics stores.

- Hot glue is actually an OK insulator against the weather
- It's difficult to see because of the camouflage

BIRDBOX

A project from the Raspberry Pi Foundation (**magpi.cc/birdbox**), this camera trap uses some concepts you can find in the other projects here, but this one involves constructing a birdbox too – a classic beginner's woodworking project.

Be careful to make sure any project like this will not disturb the birds if they nest in the box, though.

MONOFILM A600 300 MONOFILM A600

It seems the popular way to make your own wildlife camera is to house your electronics in a watertight piece of Tupperware and, apparently, use a large plastic soda bottle as a sort of weather shield for the lens of your camera. At least, this is how the My Naturewatch team like to build their cameras.

The main part of the process is building the enclosure, as My Naturewatch has a pre-built SD card image that you can load onto Raspberry Pi which will also let you connect a phone, or other Wi-Fi enabled device, so that you can control it and also see what it has taken.

If you're looking to build one, Pimoroni also sells a kit with all the electronics you'll need here: **magpi.cc/mynaturekit**.

My Naturewatch
mynaturewatch.net

- You can browse the camera with a phone or other device
- The solution is very DIY, but it works!

Antarctic PiCam
magpi.cc/130

- The hardware is attached to a wooden sled that slips into the PVC piping
- It was used in Antarctic waters to study ice algae

MONOFILM A600 300 MONOFILM A600

Fancy taking photos under the water? The Antarctic PiCam project we featured in issue 130 (**magpi.cc/130**) was created by a school student and makes use of a very simple plumbing seal, tightened by a nut, to create a waterproof enclosure. With a clear acrylic plate on one end, the camera can point outward and take footage. Software-wise, it's a little more crude in its original configuration, but by using some of the techniques in My Naturewatch, the Simple Camera Trap should allow you to make something a bit more useful for nature photography.

Alert!
Power Tools

Be mindful of problems caused by power tools. Be careful of vibration, noise hazards and wear protective clothing and goggles.

magpi.cc/powertools

Full kit

Cut to the chase with Naturebytes

Naturebytes Wildlife Cam
naturebytes.org
£110 / $137

▲ The hardware is attached to a wooden sled that slips into the PVC piping

Learn all about Astrophotography in *The MagPi* 128 (**magpi.cc/128**)

MONOFILM A600 300 MONOFILM A600

The Wildlife Cam from Naturebytes is a Raspberry Pi A+–based device that comes with a fantastic case that holds Raspberry Pi, a Raspberry Pi Camera Module, an IR lens to optimise motion detection, and even an IP55 weatherproof rating. This means it can protect against quantities of dust that would affect operations, and also from a water jet, although hopefully you wouldn't be pointing your pressure washer at it anyway.

You will need to assemble it yourself. However, it just requires screwdrivers, and the instructions are very comprehensive with plenty of photos – and there are some video guides too. In addition, you can switch out Raspberry Pi A+ for a Zero or Model B form factor, as well as other kinds of camera modules if you really want to upgrade the quality of the photos.

You can also just get the case on its own for half the price if you want to customise it further. The kit has been around for many years now and is a staple Raspberry Pi kit.

Other outdoor photography

What else can you see from your garden?

Cheap All Sky Camera
magpi.cc/cheapskycam

Using simple weatherproofing such as a plant pot and a dome, this simple build can help you track meteors in the sky using motion-sensing code. It uses a 180-degree fish eye lens to capture more of the sky, and the resulting photos have included meteors flashing across the sky and time-lapse star trails.

Star trails are a great style of astrophotography to try out

JULY IN NATURE

In the UK, July is the first full month of summer, so you'll see more creatures and critters about. This includes bats, the common lizard, osprey chicks, stoats, and many kinds of butterfly. Check out **discoverwildlife.com** for more info on the kind of things you can encounter in your garden – and beyond! - during the year.

Plant growth timelapse
magpi.cc/planttimelapse

Not only does this project your water plants when they're needed, it also keeps a time-lapse recording of their growth. This specific version is used in a greenhouse but, with some expert modifications (and weatherproofing), it could be used in a garden or even for indoor plants. It's actually quite a simple setup, but the time-lapse bit is extra fun.

The build does require a car battery, though

Credit: Andy Morffew

PuppyPi

▶ HiWonder ▶ **hiwonder.com** ▶ £445 / $559

State-of-the-art quadruped combines Raspberry Pi and ROS for powerful learning options. By **Lucy Hattersley**

SPECS

DIMENSIONS: 226 × 149 × 190 mm (length × width × height), 720 g, aluminium alloy body

COMPONENTS: Raspberry Pi 4 4GB, eight HPS-0618SG coreless servos, Raspberry Pi expansion board, 480P/8DOF camera

CONTROL METHODS: iOS/Android apps, PC application, Python API, PuppyPi V1 software, Lab tool, PS2 controller

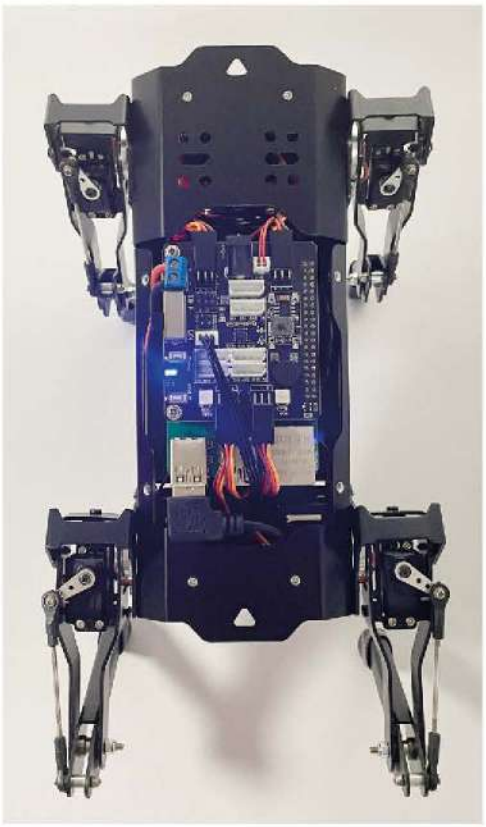
There's no denying that PuppyPi is an eye-catching robot. Inside its black aluminium alloy frame this quadruped features eight high-quality HPS-0618SG coreless servo motors with metal joints, and a six-axis IMU sensor chip MPU6050 for robot posture detection.

PuppyPi (**magpi.cc/puppypi**) is assembled by HiWonder, who created the larger SpiderPi robot (**magpi.cc/119**) and TonyPi (**magpi.cc/111**), all strong contenders in the robotics space.

At the front of PuppyPi sits a 480p camera and inside is a Raspberry Pi 4 running ROS (robot operating system), a motor control board, and 7.4V 2200mAh LiPo battery. It's a lot smaller than SpiderPi; it's cuter too, even with the black metal styling and single-eye camera. Inside the box are some coloured balls, tags, and an 8.4V charger. We also got the PS2-style controller for testing, and an Advanced kit comes with extra sensors and attachments. A pro model comes with Lidar.

Our test model worked fresh out of the box. Power it up using the switch on the side, and it instantly stands up and transmits a Wi-Fi

▶ The quadruped looks stylish with its black aluminium casing



network to join. The HiWonder iOS/Android app enables direct control of PuppyPi from a smartphone. You remotely control the robot as the camera view displays on your screen. A range of pre-defined actions are available: sit, bow, shake hands, and so on. There are also colour recognition, target colour tracking, QR code tag recognition, OpenCV face detection, and line-following modes.

You control the robot via the Android or iOS app, a desktop app, or using the optional PS2-style controller. Next, we flipped the protective lid off PuppyPi to take a closer look inside. There is a Raspberry Pi 4 with 4GB RAM and a motor control and expansion board. This features a barrel jack for charging, four servo ports, two GPIO ports, and four IIC ports, RGB lights, and two programmable buttons. Extensions offered by HiWonder include a Lidar module, ultrasonic sensor, MP3 module, display, and touch sensor.

The software running on PuppyPi is Raspberry Pi OS with custom applications and ROS (robot operating system) support. Connecting to Raspberry Pi via VNC reveals a PuppyPi desktop application that is used to visually program and chain together commands. You can dig deeper with a Python API, and it supports Gazebo simulation (**gazebosim.org**), enabling you to simulate movement in advance.

Deep training

Once you've connected PuppyPi to your network, it's possible to move beyond the iOS/Android app and dive into the world of Python programming and ROS experimentation. To this end, a huge range of tutorial PDFs are provided by HiWonder via Google Drive (**magpi.cc/puppypidocs**). These cover everything from Python, Gazebo, ROS and OpenCV vision tracking; up to stair climbing and obstacle avoidance.

> An excellent addition to any robotics course or classroom

We didn't encounter any real problems during testing. Both Raspberry Pi OS and the PuppyPi PC_Software app defaulted to Chinese (easy enough to fix in Preferences). PuppyPi is obviously not cheap in the world of Raspberry Pi robotics, but Boston Dynamic's Spot robot comes in at roughly £200,000, so it's relative. We think this would be an excellent addition to any robotics course or classroom.

The Raspberry Pi accessory board connects to the servo motors, and provides GPIO and IIC connections for accessories

Verdict

A fantastic quadruped that's easy to set up and start using. Packed with detailed documentation to explore every facet of dog-themed robotics.

9/10

The Raspberry Pi OS PuppyPi V1.0 app used to chain together commands

u-maker Box

SPECS

DIMENSIONS:
120.4 mm × 120.4 mm × 35.5 mm

WEIGHT:
93.2 g

FASTENERS:
Magnetic

▶ Weidmüller ▶ **magpi.cc/umaker** ▶ £21 / $27

Not sure what kind of case to get? Try out the case that can be just about anything. By **Rob Zwetsloot**

Choosing a case for your Raspberry Pi can be a little tricky when there are many competing products on the market. You can always make one with LEGO, 3D-printed parts, cardboard, etc. if you have the means, making the choice even more complex.

The u-maker Box may not solve that issue, but it is at the very least quite unique in its approach to Raspberry Pi case design. You can plonk a Raspberry Pi 4 inside it and be done if you want, and with magnets fastening the whole thing and plenty of space inside it, it's not the worst case you could use.

What's cool, though, is that you can really modify and personalise the case to what you need it to do thanks to a smart construction system and freely available templates to play around with for 3D printing.

▲ The standard box is very unassuming but makes for a good basic enclosure

▲ The ridges on the corners are used smartly to hold the case on various accessories

Snap build

The standard box comes with some mounts that allow you to attach it to a wall or VESA mount, and even a special one for boom arms. As well as the extra bits in the Extension Kit that help connect extra boards, brackets, or even a case fan, there's a ton of Thingiverse files for various kinds of stands, extenders, and even an alternate lid that can fit a square HyperPixel.

The number of options are truly astounding and none of it looks or feels hacky. It's just a very nice implementation of the design idea.

Verdict

A unique and very cool way to customise or prototype a case for your Raspberry Pi projects.

10/10

DISCOVER & LAUNCH NEW HARDWARE PROJECTS

BUILD YOUR OWN RASPBERRY PI-POWERED SELF-BALANCING ROBOT OR AUTOMATIC ELVEN TEA MACHINE. DISCOVER THESE PROJECTS AND MORE.

FANTASTIC FUNDING IS THE SPARK FOR DIGITAL ALCHEMISTS LIKE YOU! TURN YOUR PROJECT INTO A FULLY FUNDED PRODUCT OR SUPPORT OTHERS.

FANTASTIC FUNDING®

fantasticfunding.com

TILT
A wireless hydrometer and thermometer that connects to your Raspberry Pi
Free international shipping with checkout code:
SHIP-FREE-INTL
shop now at tilthydrometer.com

Inky Impression 7.3″

SPECS

DISPLAY:
7.3-inch e-ink, 800×480 pixels, seven colours

FEATURES:
4 × user buttons, I2C and SPI breakout pins, booster header, stand-offs

DIMENSIONS:
170.2 × 111.2 mm

▶ Pimoroni ▶ **magpi.cc/impression7** ▶ £70 / $72

Ideal for showing artwork and data, this is a super-size e-ink display. By **Phil King**

The Inky Impression 7.3″ is a new super-size version of Pimoroni's full-colour e-ink/e-paper HAT for Raspberry Pi. We reviewed the original 5.7-inch model in issue 102 (**magpi.cc/102**); there's also a 4-inch version.

> The big draw here is the size of the screen, measuring 7.3 inches diagonally

The 7.3-inch display is considerably larger than a full-size Raspberry Pi board, so is supplied with an extra female header to boost its height, along with some metal stand-offs and screws to secure it. You need to take a little care when mounting it, holding it by the sides rather than pushing down on the glass screen.

▲ On the rear you'll find a breakout header and some cool silkscreen artwork

▼ The 800×480 widescreen display can render dithered colour images with a fair amount of detail

The rear of the Inky Impression board also features a set of breakout pins for power, ground, I2C, and SPI to connect extra components such as sensors – perhaps for a weather display.

Widescreen wonder

Naturally, the big draw here is the size of the screen, measuring 7.3 inches diagonally. A resolution of 800×480 equates to a pixel density of 127.8 ppi – just marginally lower than on the 5.7″ model. In practice, images look detailed and smooth enough from a 30 cm-plus viewing distance.

To display a custom image (JPG, PNG, etc.), you can use Pimoroni's handy dithering code to process it for the seven-colour e-paper. A one-line installer is used to download the Python library, fonts, and examples from the GitHub repo (**magpi.cc/inkygit**). Note that you need I2C enabled to detect the board.

The main downside, as with all colour e-ink displays, is a slow refresh rate: around 40 seconds, depending on the complexity of the image being drawn.

Verdict

A whopper of an e-ink screen with full colour and plenty of room to display images and/or data.

8/10

WuKong 2040

Elecfreaks | magpi.cc/wukong2040 | £8 / $10

Do just about everything with a Pico thanks to WuKong 2040.

By **Rob Zwetsloot**

SPECS

SIZE:
55.8 × 87.8 × 36.3 mm, 50 g (without Pico or battery)

POWER:
18650 lithium battery

60-minute battery life

2.8 V ~ 4.2 V

INTERFACES:
2 × buttons, 2 × LEDs, 1 × buzzer, motor interface, GPIO interface, I2C interface

Verdict

Small, reasonably priced, and packed with functionality, this board is a fun way to learn and grow with Pico.

9/10

One of the more popular type of Raspberry Pi Pico add-ons that we hadn't quite foreseen was the educational breakout board. Pop your Pico onto a board and start coding it to control different components and read various sensors – it's a great educational tool for people of all ages, and usually dirt-cheap too.

At under £10, the WuKong 2040 certainly keeps prices low while still packing a huge array of features on a small board. Buzzers, buttons, LEDs, motor controllers, a traditional GPIO breakout to hook up more, and the ability to power it all with a rechargeable battery so you can take it anywhere.

The board is packed with features that make great use of Pico's own functionality

> At under £10, the WuKong 2040 certainly keeps prices low while still packing a huge array of features on a small board

Code by numbers

Programming a Pico attached to the board is quite simple, making use of standard MicroPython, CircuitPython, and C libraries to work, and each pin and component labelled with the corresponding GPIO to which it's connected. Examples are given in CircuitPython on the website, which can be fairly easy to translate to MicroPython if needed, and give you a nice little overview of what you can do with the board.

For those wanting a more practical application of the board, it also comes with a little add-on that makes the base compatible with many kind of brick building systems like LEGO – a good way to create a fun robot completely powered by custom code you made yourself.

Some great examples of how you can use the included base adapter for brick-building systems

Astradis Elektronik is Raspberry Pi's distributor for the industry

Discover the extensive possibilities of Raspberry Pi and revolutionise your projects, industrial applications and embedded systems. With its compactness, versatility and compatibility, Raspberry Pi offers a future-proof solution. You get robust computing power and extensive peripheral support, as well as unparalleled hardware and software support trough an engaged community worldwide.

With us, you can buy Raspberry Pi's innovative products and rely on reliable and timely deliveries. Our many years of experience in logistics and warehousing enable us to offer you many ways to optimise your production process and thus reduce costs.

Time to Technology

- *In addition to Raspberry Pi products , **we offer** you many suitable components from leading manufacturers, **we optimise** your parts list and combine components into assemblies, which we then manufacture and deliver for you.*
- *In addition, **we provide** you with development tools for Raspberry Pi products that we have created.*

  These include, for example, HATs for mobile communication, displays that plug- and-play Raspberry Pi drivers, and development boards with a variety of interfaces to exploit the full potential of Raspberry Pi.

ATXX 5G Shield HAT

Time to Market

- *To successfully bring your product to market, **we help you optimise the design** and **find the best solution for distribution**.*
- ***We offer flexible external storage solutions** from micro SD cards to integrated SSD hard drives.*
- *Together with our partners, **we can securely flash your software and meet your security requirements**.*
- ***We are happy to organise production for you** and **deliver a finished ODM product with the „Powered by Raspberry Pi" label**.*

Do you have an idea for partnership? Please contact us @ a.feix@atxx.de; Phone: +49 89 89 505-231

Badger 2040 W (Pico W Aboard)

SPECS

DISPLAY:
2.9-inch e-ink, 296×128, monochrome

FEATURES:
Pico W, 5 × user buttons, reset button, LEDs, Qwiic/STEMMA QT port

DIMENSIONS:
85.6 × 48.7 × 10 mm

▶ Pimoroni ▶ **magpi.cc/badger2040w** ▶ £27 / $28

Much more than a badge, this versatile mini e-ink display has wireless connectivity. By **Phil King**

▼ On the rear you'll find Pico W, a reset button, battery connector, and Qwiic/STEMMA QT port – to which you could connect a sensor

The Badger 2040 W is a mini monochrome e-ink display with wireless connectivity. Unlike the original Badger 2040 (reviewed in issue 116, **magpi.cc/116**), this new model is based around a Pico W pre-soldered to the rear.

Another improvement is the inclusion of a real-time clock (RTC), enabling it to keep time even when in 'super deep sleep' mode, with Pico W powered down – in which it draws a minuscule 20 µA of current.

Talking of power, you will want a portable way of supplying it. To this end, there's a JST battery connector on the rear. If you buy the accessory kit version, you get – along with a lanyard – a 2×AA battery pack with a Velcro patch to stick it to the rear of the board.

Alternatively, you could use a standard USB power bank connected to Pico W. Both solutions are a little chunky, though, so you might prefer a slimline LiPo battery pack.

▲ The Badger 2040 W's mini monochrome e-ink display is surrounded by tactile buttons

> A real-time clock enables it to keep time even when in 'super deep sleep' mode

Easy to program

When first powered up, the Badger 2040 W launches into 'Badger OS' with a scrollable menu of icons to choose demo programs and tools. These include a badge, digital clock, e-book reader, interactive checklist, news headlines, and weather dashboard.

Naturally, you can connect Pico W to a computer via USB to customise the examples or create new programs – in MicroPython or C/C++. Pimoroni's standard PicoGraphics library makes it easy to add bitmap images, and several fonts are supported. You'll need to add your Wi-Fi credentials to try out examples such as news and weather, as well as setting the correct time via NTP.

Verdict

An interactive badge that doubles as a versatile mini e-ink display with a reasonably quick refresh rate and Wi-Fi connectivity.

8/10

10 Amazing:

Cases

Protect your Raspberry Pi with these excellent enclosures

Raspberry Pi is hardly fragile, but it really is recommended to get some kind of case for your microcomputer (or microcontroller), even if you're just cutting some holes in the cardboard box it comes with. Here are ten great cases that are a little more sturdy though.

SmartiPi Touch Pro

Case, stand, display

With an official touch screen and a Raspberry Pi, you can create a quick interactive display that needs very little space and is extremely customisable.

magpi.cc/smartipipro | £30

FLIRC case

Media shelf ready

Designed for use by your TV, this range of cases doesn't look out of place next to Blu-ray players or satellite boxes.

magpi.cc/flirc | £12

Argon EON NAS

File server

The price may be eye-watering, but this enclosure allows you to add four hard drives, includes an RTC, cooling fan, programmable OLED display, and room for your Raspberry Pi.

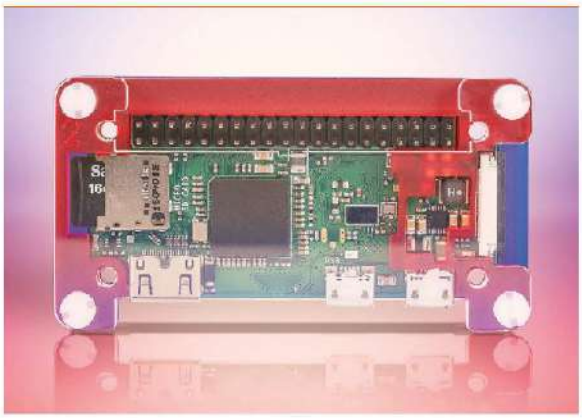
magpi.cc/argoneon | £144

Pibow

Layers of case

Pimoroni's original flagship product is this sturdy case made up of laser-cut acrylic layers. There are many styles and colours to get as well.

magpi.cc/pibow | £6



SecurePi Case

Locked down

This case allows you to cover up IO ports on Raspberry Pi, including the SD card slot, for increased security.

magpi.cc/securepi | £10

PiBoy XRS

Handheld alternative

We gave this case a 10/10 a few months ago in *The MagPi* as it's pretty much the ultimate retro handheld machine. You can even stream Steam games to it.

magpi.cc/piboyxrs | £120 / $150

Official case

Simple and clean

The official case snugly fits a Raspberry Pi inside. There's a version of all types of Raspberry Pi, and even a case fan if you're using your Raspberry Pi at full power regularly.

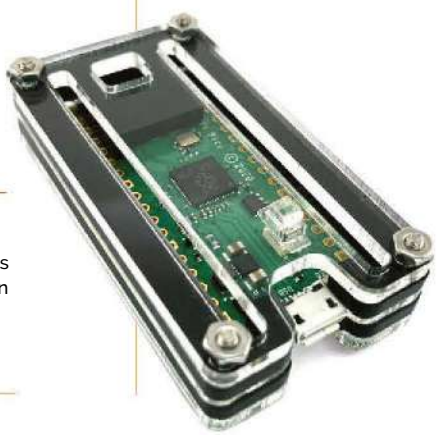
magpi.cc/case | £6

NESPi 4 case

Classic console body

Lots of people like to use a Raspberry Pi for retro gaming, so what better way to display it than with it inside a retro console case with a fake cartridge?

magpi.cc/nespi4 | £29



Zebra Pico case

Small layers

Due to the way Pico is used as a microcontroller inside projects, you rarely see cases for it. It still looks great in one if you're using it for prototyping though.

magpi.cc/zebrapico | £7 / $9

Learn Assembly with Raspberry Pi

Get to grips with the obtuse technology that underpins all modern computers

Introduction to Computer Organization: ARM Assembly Language Using the Raspberry Pi
Robert G. Plantz
Front Matter
1 Introduction
2 Data Storage Formats
3 Computer Arithmetic
4 Basic Data Types
5 Boolean Algebra
6 Logic Gates
7 Logic Circuits
8 Central Processing Unit
9 Programming in Assembly Language
10 Structure of the Intel Pentium

Programming from the ground up

AUTHOR
Robert G. Plantz

Price: FREE

magpi.cc/introarm

Assembly language is the low-level coding language used to control computers.

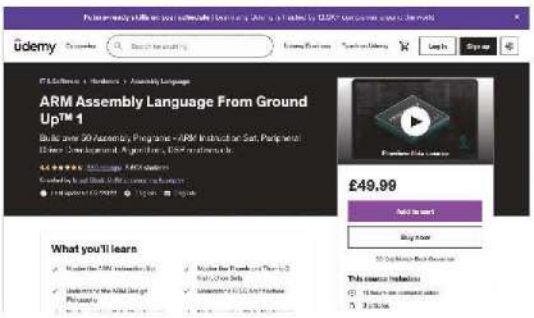
When you program in a higher-level language, like C or Python, it is converted into Assembly so that the computer can run it.

Assembly is nearly one-to-one with machine code, the series of instructions made up of zeros and ones that make up the data and logic commands that all computers run. It can be baffling, but learning (or at least attempting to learn Assembly) is a noble pursuit for all coders and computer users.

This website, Introduction to Computer Organization: ARM Assembly Language Using the Raspberry Pi, is a resourceful guide to learning ARM Assembly. The online book provides a comprehensive learning path to understanding computer organization and how it relates to ARM Assembly language. With detailed sections on programming, addressing modes, data movement instructions, arithmetic, logic instructions, and more, this guide ensures a strong foundation for anyone keen on understanding the nuances of ARM Assembly language and computer organization.

ARM Assembly courses

Learn ARM Assembly by following a course online



ARM ASSEMBLY LANGUAGE FROM GROUND UP

This Udemy course, by Israel Gbati, is well-structured and covers the fundamentals and advanced concepts of ARM Assembly language, making it ideal for beginners and intermediate learners.

magpi.cc/udemyarm

EMBEDDED SYSTEMS WITH ARM

This Coursera course, provided by ARM, dives deep into Assembly language programming, focusing on ARM Cortex-M microcontrollers.

magpi.cc/courseracortexm

ARMV8-M ARCHITECTURE FUNDAMENTALS

This course focuses specifically on the Armv8-M version of the Arm Architecture, used in Raspberry Pi 4. It's another one created by ARM and describes the instruction set and architecture in detail.

magpi.cc/courseraarmv8

Raspberry Pi Assembly Language Programming

AUTHOR
Stephen Smith

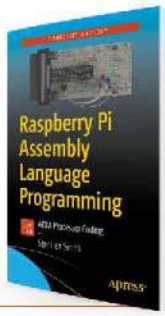
Price:
£35/$35
magpi.cc/rpiassembly

Stephen Smith is a high-profile name in the world of ARM Assembly language, so much so that we asked him to write us a series of Assembly tutorials in *The MagPi* (starting with issue 116, **magpi.cc/116**).

His book, *Raspberry Pi Assembly Language Programming*, is the one to get for anyone interested in delving into the depths of Assembly language using Raspberry Pi.

This book takes a focused approach to teaching Assembly language in an accessible and engaging way, utilizing the versatile Raspberry Pi for practical, hands-on learning experiences.

Readers are guided through the fundamentals of Assembly language, and then onto more complex programming concepts. The journey culminates in the development of a fully functional bare metal operating system.



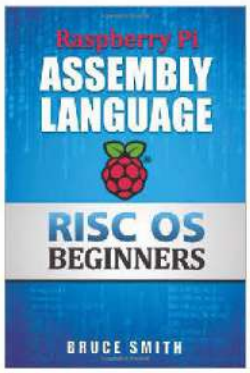
Raspberry Pi Assembly Language RISC OS Beginners

AUTHOR
Bruce Smith

Price:
£16/$16
magpi.cc/riscosassembly

RISC OS is an alternative operating system originally designed by Acorn computers and kept alive by volunteers.

Bruce Smith uses RISC OS to provide a unique and insightful guide to the fundamentals of Assembly language programming, specifically for the Raspberry Pi.



The book starts with an introduction to Raspberry Pi and RISC OS, setting the stage for a comprehensive exploration of Assembly language programming. Each chapter building upon the last, covering topics from basic concepts to more advanced aspects of Assembly language.

The book's focus on RISC OS provides a different perspective on Raspberry Pi programming, which makes a lot more sense when you are so deep in the world of ARM Assembly. While RISC OS and ARM Assembly are both niche subjects, they tag team well together.

ARM Assembly videos

Understanding ARM by following these YouTube courses

ASSEMBLY LANGUAGE TUTORIAL

Derek Banas is known for his clear and thorough style, making complex topics accessible to beginners. This video serves as a great introduction to Assembly with Raspberry Pi.

▶ magpi.cc/assemblytutYT

ASSEMBLY LANGUAGE PROGRAMMING WITH ARM

This extensive video tutorial from freeCodeCamp covers ARM Assembly language with ARMv7 from the basics to more complex concepts.

▶ magpi.cc/freeCCARM

INTRODUCTION TO ASSEMBLY PROGRAMMING WITH ARM

This series of 16 tutorials by OliveStem demonstrates ARM memory layout, arithmetic, and logical operations.

▶ magpi.cc/OliveStemARM

Alistair MacDonald

Maker and Maker Space founder, Alistair likes to bring the joy of his hobby to the community

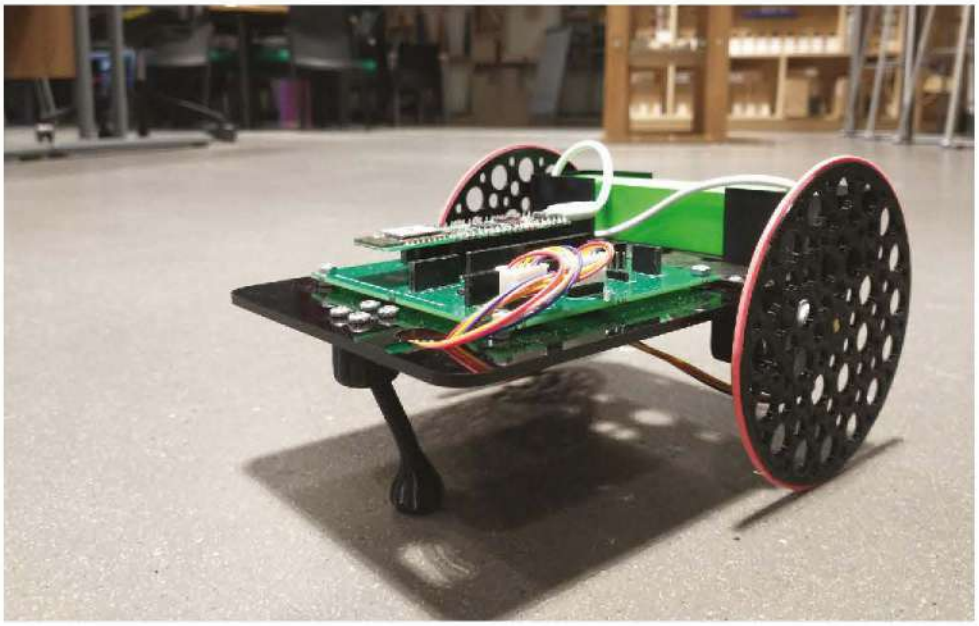
- Name Alistair MacDonald | Occupation FabLab manager
- Community role Event organiser | URL fablabsunderland.org

A popular plot device for stories is simply 'I'm bored of my job and I am going to do something else'. It's popular because it's relatable, even if not everyone is able to manage it. Alistair MacDonald did though after spending years developing "big boring software."

"I needed a change, so I set up on my own with the aim of doing more fun and interesting projects," Alistair tells us. "This worked out, and I gained a reputation and most of my work came through word of mouth. In the end, I specialised in prototyping products, creating interactive displays, and supporting artists in creating artworks on the side. It was random, and at times just crazy, and I liked that."

He began to teach others his skills at a local university, and this has expanded even further to becoming a founding member of the Newcastle hackspace called Maker Space.

"Today I run FabLab Sunderland where we support both students and businesses alike in creating almost anything," Alistair says. "We offer 3D printing, laser cutting, electronics fabrication, CNC routing, vinyl cutting, and many, many other things."



Trundlebot is an educational MicroPython robot created by Alistair, and it won Electromaker's PCB project design contest

What is your history with making?

I don't have a memory of when I started making things. It's just what I have always done. When I was young, my favourite books were all making books. That, and books on how things worked. The earliest project I can remember is Roby, a cardboard robot with lights on the front. These were switched on and off by moving paper clips that were drawing-pinned to a piece of wood. I must have been around seven years old at the time. My skills have moved on since then but, if I am honest, my childlike joy of such things has not.

As I got older, I would study electronics catalogues and datasheets and would want to try everything. I would ask my father to buy components and make things like disco light controllers, audio mixers, and model railway control systems.

FabLab Sunderland is a great place to learn how to make, and also just make stuff yourself

Some things worked, some things did not, but I learnt from all of them.

After I graduated [university], I started doing a few small projects again, but it was not until the birth of the Arduino that I really got back into electronics. It was not until much later, when we created Maker Space, that I had access to the facilities to do more physical builds again.

> My good friend Brian and I decided to form a hackspace

When did you learn about Raspberry Pi?

Very early on. I cannot remember exactly when I heard about it, but it was long before they were publicly available. I tried to get one of the first batch and managed to place an order, but something went wrong with the order and the order was cancelled… I did still manage to get one early on though. I joke that I gained a lot of friends because of this, but a lot of people did borrow it. We have an active software development community in Newcastle upon Tyne, and a few big companies wanted to check if their software would run in the [Raspberry] Pi.

How did Maker Space start?

A long, long time ago I wanted to create a space where we could make things. Or, to be more specific, I wanted the space, and the only way it was going to exist [was] if I created it. Back then, I envisaged a kind of workshop, an art studio, and community centre all wrapped into one. In the end, my good friend Brian and I decided to form a hackspace, and we managed to rent a small space at the back of a community cinema in the city.

The problem was that, back then, no one else was looking for something like this, or if they were then they were not finding us. In the end, we realised that we needed to change how we were reaching out to people.

We did things like blogging about what we were up to. We talked at local tech user groups and events at our local science centre. We rebranded from Hackspace Newcastle to Maker Space (a unique and more descriptive name back then). We set up an online discussion group about making to try to bring people with that making DNA in them together. In the end we got more people involved and, when you have more people sharing the load, things become easier. We owe a lot to those who put the effort in today.

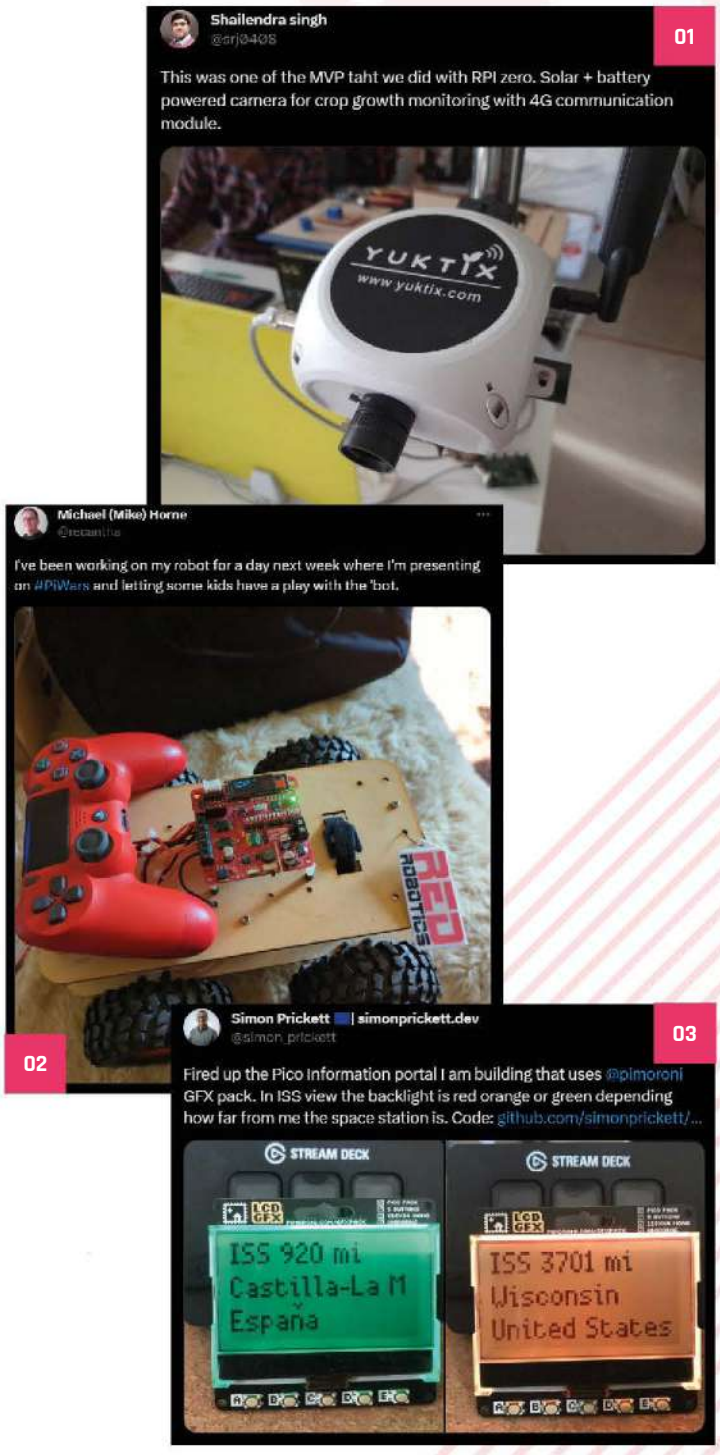
They also get into the spirit of the maker season at FabLab Sunderland

MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they have made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!


Shailendra singh
@srj0408
01
This was one of the MVP taht we did with RPI zero. Solar + battery powered camera for crop growth monitoring with 4G communication module.
YUKTIX
www.yuktix.com
Michael (Mike) Horne
I've been working on my robot for a day next week where I'm presenting on #PiWars and letting some kids have a play with the 'bot.
02
Simon Prickett | simonprickett.dev
@simon_prickett
03
Fired up the Pico Information portal I am building that uses @pimoroni GFX pack. In ISS view the backlight is red orange or green depending how far from me the space station is. Code: github.com/simonprickett/...
STREAM DECK
ISS 920 mi
Castilla-La M
España
STREAM DECK
ISS 3701 mi
Wisconsin
United States

01. A fancy piece of kit that seems great for crop monitoring

02. Pi Wars is back in a big way

03. It's very optimistic to have it be green when it's close by, even though it's 250 miles above ground

04. You too can get our books delivered around the world from **magpi.cc/store**

05. Ah, we need a few Unfinished Projects months ourselves. Excellent idea

06. A nice bit of hacking and upcycling – we like it

07. More data is always better

08. This interactive pal looks like it's going to eat our 10p coins

09. ISS tracking on little screens is popular! The API is quite readily available after all

04

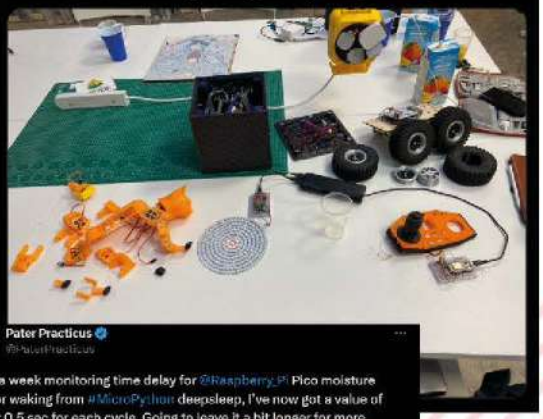
Roland Schulz 🇩🇪⚡r_schulz_maker@mastodon.social
@r_schulz_maker

Hi #MagPiMonday, i haven't been to the @Raspberry Pi popup store in Leeds. I got a package from 🇬🇧! Books full of inspirations for my next projects. I'm diving now into the world of #python literature. Thanks Raspberry Pi Press.
I playing with my @picobricks too🤖👍.

05

Dr Footleg - Roboteer @drfootleg@fosstodon.org
@drfootleg

I hosted the first 'Unfinished Projects' evening at @cammakespace last week. An event to show and tell about projects in progress, or stalled and needing fresh ideas of just some motivation to get back to them. Here are some of the things I brought along. #MagPiMonday



06

Ulli Sundermann
@ullisun68

Hi on this sunny #magPiMonday
I found in the toys box of my children this 4 wheel driven gear and in my sparepart box theese servos / sensors and a Pico. Now it is time to create a 4wheel driven robot as POC of a 4WD lane mower
#Robotics MINI #PiMowBot @TGD Consulting

Pater Practicus
@PaterPracticus

After a week monitoring time delay for @Raspberry_Pi Pico moisture sensor waking from #MicroPython deepsleep, I've now got a value of about 0.5 sec for each cycle. Going to leave it a bit longer for more accuracy before reprogramming. youtu.be/-UjNNcQtsy4 #MagPiMonday

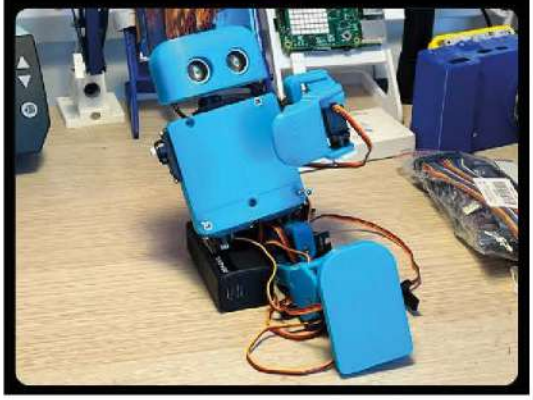
07

08

Kevin McAleer 🤖 Robot Maker
@kevsmac

Happy #MagPiMonday! This weekend I've been working on Chip - the Cute Humanoid Interactive Pal! It's powered by the RP2040 based @pimoroni Servo 2040, and #micropython obvs.

Read more about it at kevsrobots.com 🤖



09

Kiting Free
@KitingFree

My latest Pico project: an e-paper info-screen for all the essential data flying around (ISS location, weather forecast, CO2 level, Moon phase, Strava perf, etc.) raspicolas.wordpress.com/2023/06/07/tur...

Events in pictures: Raspberry Jam WV

Community and official events in the wild

Way back in issue 79 of *The MagPi*, we spoke to Brett White about how he has been trying to get more computer science education into his home state of West Virginia in the US. Since then, he's been putting on Raspberry Jams and is now a professor at the University of Charleston's Computer Science department. In March this year, the fifth Raspberry Jam WV went ahead – here are some photos.

FIND OUT ABOUT NEXT MONTH'S EVENTS AND POP-UP ON PAGE 92

01. Brett organises the Raspberry Jams, with 2022's being so good he became a professor
02. Programming and hacking Minecraft remains a popular Jam activity
03. Various kinds of making equipment were on display
04. You could also buy all manner of components for your own projects
05. What kind of projects indeed – we have several in this mag you can try

Crowdfund **this!** Raspberry Pi projects you can crowdfund this month

PiBeam

"Step into the future with us and witness the birth of a device that seamlessly merges the unrivalled computing power of Raspberry Pi's RP2040 microcontroller with the wireless capabilities of an IR transceiver. PiBeam is not just an ordinary gadget; it's a technological marvel, enabling you to explore the endless possibilities of Python-based hardware learning."

▶ **kck.st/43klhJi**

Make-It-Move HAT

"Make-It-Move HAT is the most user-friendly plug-n-play custom hardware, that eases the development of motor driven Robotics Applications. It provides an easy plug-n-play support to control up to 23 motors for any robotics Application, with easy to understand simplified code and Libraries in multiple Languages."

▶ **kck.st/43DPAKB**

Together we can make a **difference**

Give young people the opportunity to learn about technology

The Raspberry Pi Foundation enables young people to realise their full potential through the power of digital technologies, but we can't do this work without your help. Your support helps us give young people the opportunities they need in today's world. Together we can offer thousands more young people across the globe the chance to learn to create with digital technologies.

Generous donations from organisations and individuals who share our mission make our work possible.

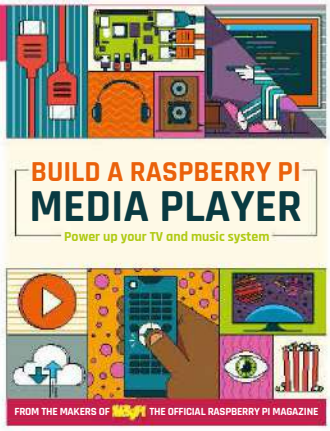
DONATE NOW

Donate today to make a difference: **rpf.io/magpidonate**

Raspberry Pi Foundation

Your Letters

Centre of media


BUILD A RASPBERRY PI
MEDIA PLAYER
Power up your TV and music system
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

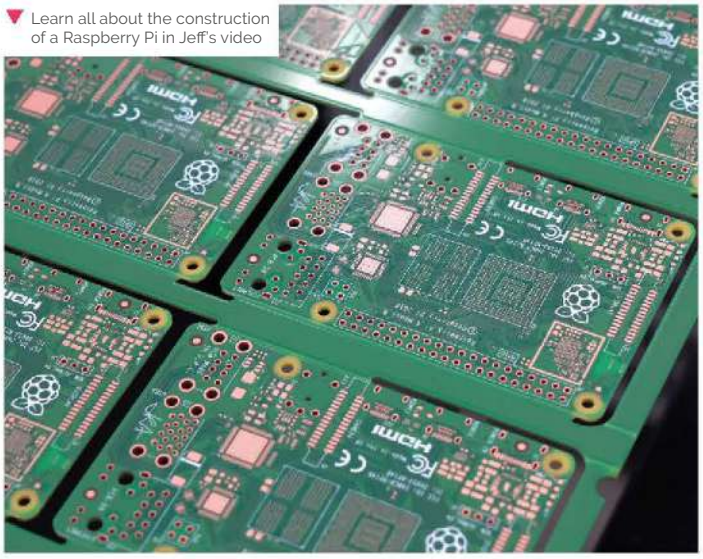
There are many projects in here that will help improve how you watch and listen

I'm considering using Raspberry Pi for a home media centre setup. Are there any step-by-step guides or tutorials available that cover the process of setting up a media centre using Raspberry Pi? I'm particularly interested in options for streaming media, managing a media library, and integrating with popular streaming platforms. Any recommendations for hardware and software configurations would be greatly appreciated.

Sean via email

We have a fantastic free PDF called 'Build a Raspberry Pi Media Player' that includes some of our most comprehensive tutorials for media centres, sound systems, retro gaming machines, and more. It even has tips on how to use streaming services as well! Find it at **magpi.cc/mediaplayer**.

Learn all about the construction of a Raspberry Pi in Jeff's video



Grand tour

I saw this video of a tour of the factory in Wales where Raspberry Pi is made and it was fascinating – when is the next available factory tour, so people can see where the magic happens?

Kyle via Twitter

Unfortunately the video that Jeff Geerling made (which you can find here: **magpi.cc/tour**) was done under special arrangement by Sony, so it's not something everyone can do. The tour is pretty comprehensive though, so living vicariously through Jeff is not the worst idea.

APL of my eye

This is a note for Lucy Hattersley about her article called 'A retro world tour' [Ed– Last month's Final Word piece]. It was a great article about the early days of personal computing. In it, Lucy mentioned that she had never heard of APL. APL is a fantastic programming language that was created in the early 1960s... APL is currently available from a company called Dyalog at no cost for personal use.

Jim via email

APL is a very interesting style of code and it's good to know it's still around and being used! We may have to do a tutorial with it in the future.

Contact us!

- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.com**
- Online **forums.raspberrypi.com**

USA SPECIAL!
6 ISSUES
FOR $43

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Python Robots
Build dynamic bots with arms and legs
CREATE A RASPBERRY PI WILDLIFE CAMERA

FREE
RASPBERRY PI
PICO W

Subscribe online:
magpi.cc/subscribe

Subscription is for the next six issues by continous credit card and renews automatically. This is a limited offer. Offer subject to change or withdrawal at any time.

Community Events Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Cornwall Tech Jam

- **Saturday 1 July**
- **FibreHub, Pool, UK**
- **magpi.cc/ctj131**

Cornwall Tech Jams are run by volunteers working in IT and education throughout Cornwall. They are supported by Software Cornwall, its members, and other local businesses. Our volunteers give their own time and expertise to plan each Cornwall Tech Jam and to the regular maintenance of all our equipment.

02. Melbourne Raspberry Pi Meetup

- **Sunday 2 July**
- **Docklands Makerspace and Library, Melbourne, AUS**
- **magpi.cc/mrpm130**

This meetup is open to everyone with an interest in electronics, robotics, home automation, 3D printing, laser cutting, amateur radio, high altitude balloons, space tech, etc. Makers are invited to bring along their projects and project ideas, and come connect with other makers. Get your questions answered, show off the work you are doing, and get support to resolve nagging issues.

03. Southend Raspberry Jam Maker Meetup

- **Saturday 15 July**
- **Southend Library, Southend-on-Sea, UK**
- **magpi.cc/srjmm131**

This Raspberry Jam community event is hosted at Southend Library and they're looking to organise more regular activities such as workshops, project demos, and larger meetups. Join the group and learn about opportunities to build technology projects, help at or run workshops, or build alongside other makers and companies.

04. Newark and LATeRe at FIEE

- **Tuesday 18 July to Friday 21 July**
- **São Paulo Expo, São Paulo, Brazil**
- **magpi.cc/fiee**

LATeRe, Brazil's exclusive representative for Newark Electronics, will be exhibiting at FIEE in São Paulo in July. They'll be showcasing Raspberry Pi 4 computers, Raspberry Pi Compute Modules, RP2040 microcontroller chips, and Raspberry Pi Pico series microcontrollers. You'll find them at booth F38. FIEE registration is required.

FULL CALENDAR

Get a full list of upcoming community events here:

magpi.cc/events

RASPBERRY PI STORE POP-UP

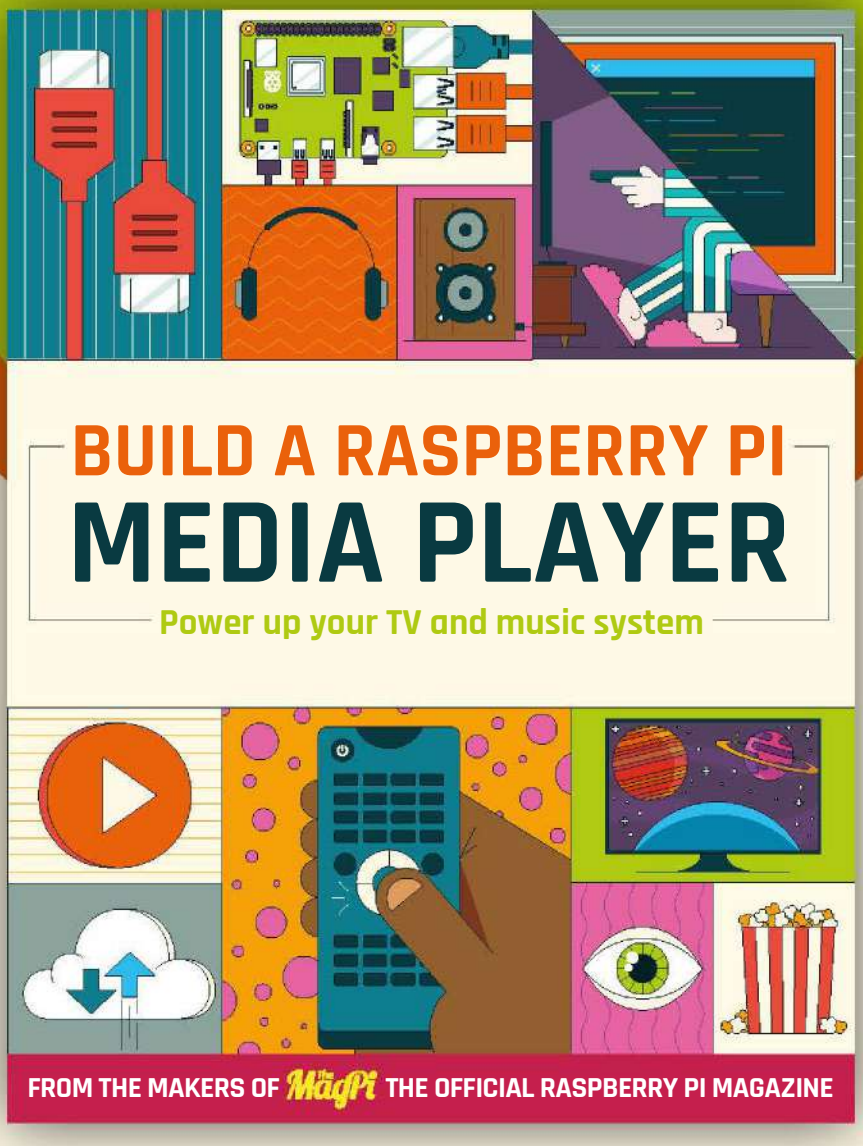
Where can you find Raspberry Pi next?

- Next location Raspberry Pi Store pop-up, Gateshead
- Where Metrocentre, Gateshead, UK
- When Friday 28 July and Saturday 29 July

At this Raspberry Pi Store pop-up in Gateshead, you can experience and buy Raspberry Pi products. Explore some of the things you can do with a Raspberry Pi, discover our accessories and books, and get your hands on limited edition exclusives.

magpi.cc/popup2023gateshead

Your FREE guide to making a smart TV


BUILD A RASPBERRY PI
MEDIA PLAYER
Power up your TV and music system
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

magpi.cc/mediaplayer

WIN ONE OF TWO ARGON POD SETS IN ASSOCIATION WITH ARGON40

The Argon POD system is a modular case for Raspberry Pi Zero that allows you to build up and add functionality, depending on your needs. As well as adding Ethernet support, you can also convert the micro USB and mini HDMI to their full-size alternatives, and we have two complete sets to give away.

Head here to enter: magpi.cc/win
Learn more: magpi.cc/argonpod

Terms & Conditions

Competition opens on **28 June 2023** and closes on **27 July 2023**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter or any other companies used to promote the service.

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE **#68**

OUT NOW

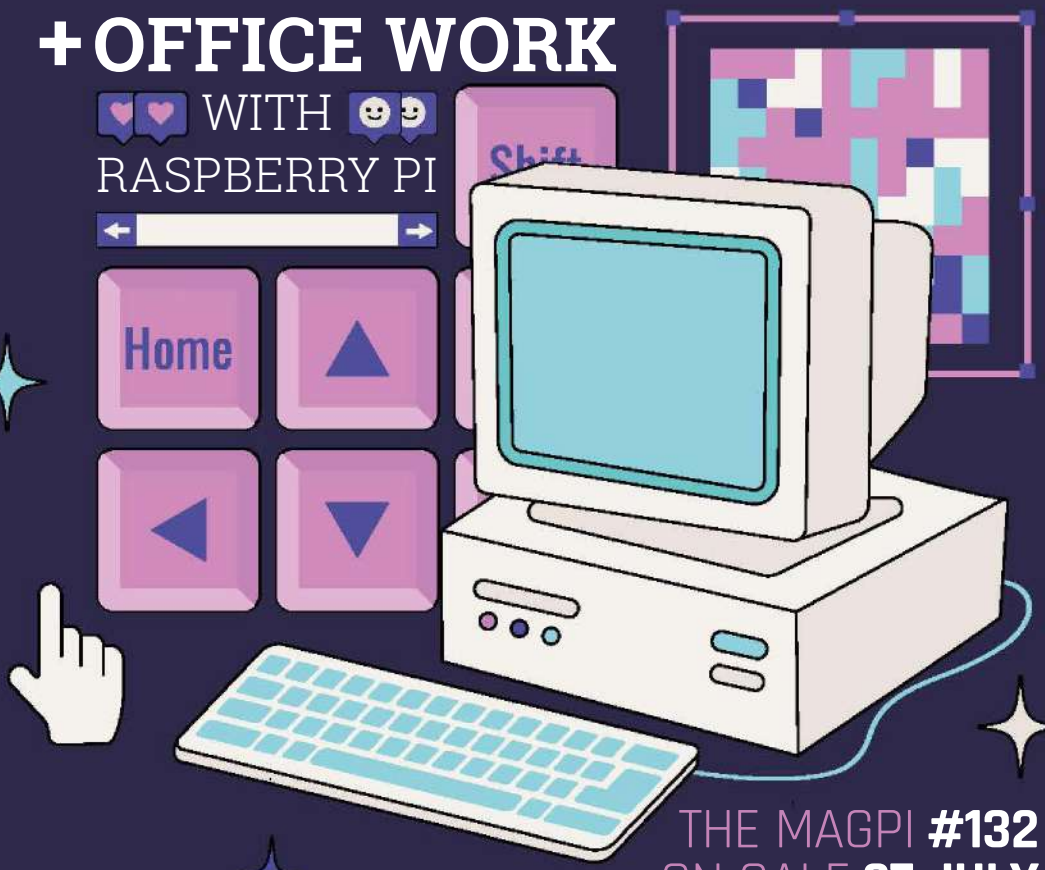
hsmag.cc

Available on the App Store

GET IT ON Google Play

RETRO GAMING

+ OFFICE WORK WITH RASPBERRY PI



THE MAGPI **#132**
ON SALE **27 JULY**

Plus!

Star Wars diorama

Paragraphica AI lens-free camera

Pipstrelle Bat Detector

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER @TheMagPi
FACEBOOK fb.com/MagPiMagazine
EMAIL magpi@raspberrypi.com

EDITORIAL
Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Nicola King

ADVERTISING
Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Olivia Mitchell, Sam Ribbits

Illustrator
Sam Alder

CONTRIBUTORS
David Crookes, Rosemary Hattersley, Nicola King, Phil King, Dr. Andrew Lewis, KG Orphanides, David Plowman, Danny Staple, Stewart Watkiss

PUBLISHING
Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.

Tech in front of our eyes

Virtual reality is in the news, but physical making is the real deal!

By Lucy Hattersley

The future is amongst us. At least according to Apple, purveyor of shiny slabs, which recently announced a new glass headset to slap in front of your face. You can dial back reality using a rotating tool on the side, and tap apps and icons in mid-air. Sounds fun.

I'm not convinced by VR or AR. Does the future of technology really belong to this kind of detachment from reality? Who knows! Maybe I'll have egg on my face and a Vision Pro headset in three years. I'll be writing this with giant hovering buttons in front of my face while you read it, in thirty-foot-high letters being beamed into your retinas from an inch away.

I'll bet not though. I'll be happily playing with Raspberry Pi computers and building ever more advanced robots and projects.

Still, there's a lot going on in the tech industry at the moment. From strident leaps in AI (speaking of which, have you seen GitHub Copilot?) through to shiny new Apple toys, it feels like the tech industry is getting ready to limber back up.

Raspberry Pi itself is shaking off the effects of the supply chain problems that have plagued everybody for the last couple of years. There is no steady supply of Raspberry Pi Zero, Raspberry Pi 3A+, and an increasing supply of Raspberry Pi 4 models.

Back in the room

This comes as a big relief to us. We're keenly aware that people love this magazine because they love Raspberry Pi. And it's easier to love something that you can reliably buy.

> Maybe, when I'm hooking up dual tiny screens inside a 3D-printed headset, it'll suddenly feel real to venture into an open-source virtual world

This brings us back to this weird virtual world that Apple seems to be pushing for. I don't believe that real tech people want to point and tap into empty space. For me, the joy of computing is linked to the real world. It's physical things you can pick up, hold, and use. Actual computers with wires and buttons and solder. Building an arcade machine, robot, or earthquake detector. It doesn't matter if it's a serious project or a piece of whimsy. What matters most is that it's real.

The future appears one day and then suddenly it's the present. It feels like ChatGPT and the AI generative genie are out of the bottle and will be staying with us, for now. I suspect we'll also see a lot more generated AI art and 3D visuals. Personally, I hope that it'll be the mundane but useful technology, the sort that pings me for calendar events and lets me know when my partner is calling, but then fades into the background when I'm looking at more interesting things.

Maybe VR and AR are the future, but I need to assemble it with my own two hands for it to feel real. There have been some attempts at VR with Raspberry Pi, and with its dual HDMI output, there is potential for VR projects with our little board computer. Maybe, when I'm hooking up dual tiny screens inside a 3D-printed headset, it'll suddenly feel real to venture into an open-source virtual world. Until then, I'll be sticking with actual reality.

AUTHOR

Lucy Hattersley

Lucy is editor of *The MagPi* and is currently causing an office kerfuffle that she can't talk about. Maybe that VR escape isn't such a bad idea after all.

magpi.cc/lucyhattersley

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiShop.us PiShop.ca PiShop.mx ThePiHut Pi-Shop.ch Welectron.

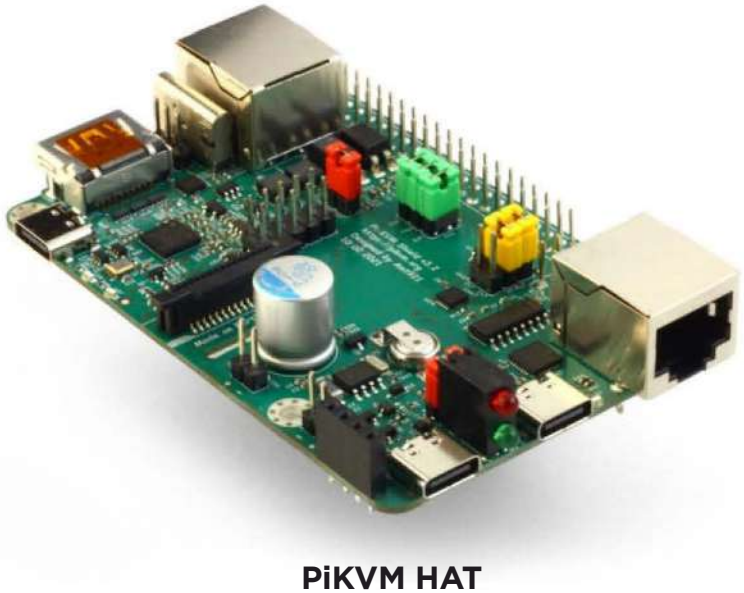
Contact your favorite Pi store if it's not listed here

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects

Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io